



Quantum-Safe Encryption Using Postquantum Preshared Keys

This chapter explains how to use Postquantum Preshared Keys (PPK) for quantum-safe encryption of IKEv2 and OTNsec data, through the implementation of RFC 8784 and the Cisco Secure Key Integration Protocol (SKIP).

- [Quantum-Safe Encryption Using Postquantum Preshared Keys, on page 2](#)
- [Verify the PPK Configuration, on page 10](#)
- [View IKEv2 Session Detail, on page 14](#)

Quantum-Safe Encryption Using Postquantum Preshared Keys

Table 1: Feature History

Feature Name	Release Information	Description
SKIP Protocol Support for Quantum Safe IKEv2 Encryption	Cisco IOS XR Release 24.1.1	Traditionally, the IKEv2 encryption was vulnerable to quantum attacks. Now, IKEv2 encryption complies with RFC 8784, which specifies using postquantum preshared keys (PPK) to make it resilient to quantum attacks. You can generate both manual and dynamic PPKs. The dynamic PPKs are generated using the Cisco Secure Key Integration Protocol (SKIP). The IKEv2 encryption is configured through CLI or by the Cisco-IOS-XR-um-ikev2-cfg Yang model. CLI: • The ppk manual/dynamic keyword is introduced in the keyring command. • The keyring ppk keyword is introduced in the ikev2 profile command. • The sks profile command is introduced.

Quantum computers have raised concerns about the security of cryptographic algorithms used extensively today. AN example of a cryptosystem that could be vulnerable to quantum computers is the Internet Key Exchange Protocol Version 2 (IKEv2). Any VPN communications could be decrypted in the future when a quantum computer becomes available. To address this issue, IKEv2 can be extended that uses preshared keys making it resistant to quantum computers.

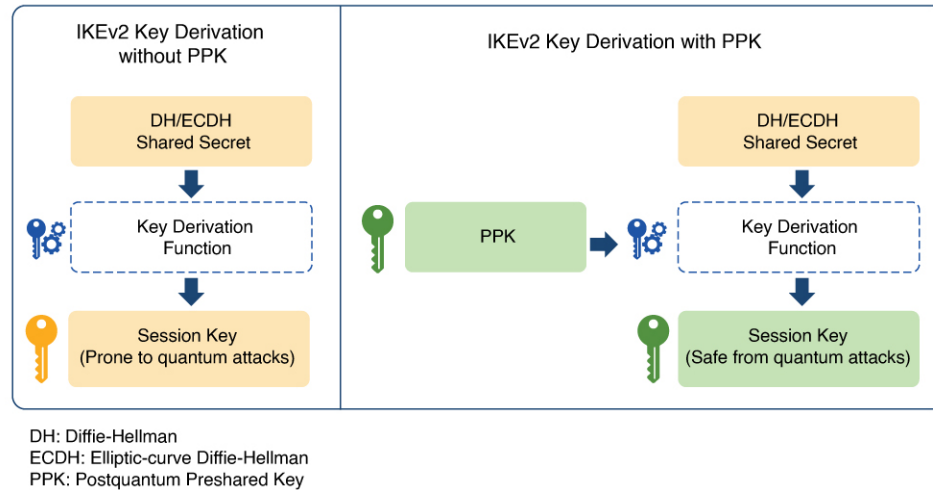
Postquantum Preshared Keys

Session keys that are derived from preshared keys are safe to quantum attacks if the preshared keys are endowed with sufficient entropy. Therefore, the resulting system is deemed secure against classical attackers of today, and against future quantum attackers.

RFC 8784 (Mixing Preshared Keys in IKEv2 for Postquantum Security) outlines an enhancement to the IKEv2 protocol that renders it quantum-computer-resistant through the incorporation of preshared keys, referred to as PPKs. This RFC establishes the specifications for enabling PPK capability negotiation, PPK ID transmission,

the integration of PPK as a supplementary factor in session key derivation, and the potential fallback to sessions not dependent on PPKs.

Figure 1: IKEv2 Key Derivation - With and Without PPK



Dynamic Postquantum Preshared Keys and SKIP

The Cisco Secure Key Integration Protocol (SKIP) is an HTTPS-driven protocol designed to enable encryption devices like routers to import PPKs from an external key source. These externally imported PPKs, referred to as dynamic PPKs, provide advantages such as automated provisioning and updates, and improved PPK entropy. Encryption devices must have the SKIP client and the external key source must have the SKIP server.

To be SKIP-compliant, an external key source must follow the Cisco SKIP protocol and use an out-of-band synchronization mechanism. This ensures that the same PPK is provided to both the initiator and responder encryption devices. The external key source can be in the form of a Quantum Key Distribution (QKD) device, software, or cloud-based key source or service.

The external key source must meet the following expectations to be SKIP-compliant:

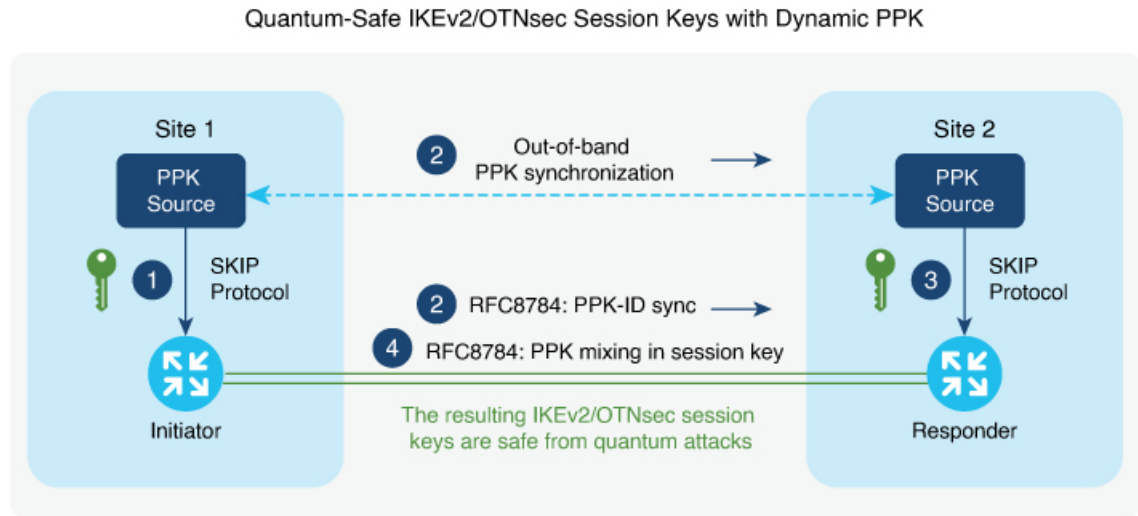
- Must implement the SKIP protocol or API, as specified in the Cisco SKIP specification.
- Must provide the same PPK to the encryption device pair—initiator and responder—using an out-of-band synchronization mechanism.



Note Key source vendors, such as QKD vendors, must contact their Cisco representative to implement the Cisco SKIP protocol.

The following figure shows quantum-safe IKEv2 and OTNsec session keys using dynamic PPK.

Figure 2: Quantum-Safe IKEv2 and OTNsec Session Keys with Dynamic PPK



The IKEv2 initiator and responder are connected to their respective local key sources and are configured with the SKIP client that specifies the IP address and port of the key source. The PPK sources are also configured with the SKIP parameters, which include the local key source identity and a list of identities of the peer key sources.

The high-level operation of Cisco SKIP protocol is as follows:

1. The IKEv2 initiator places a request for a PPK from its key source. The key source replies with a PPK and the corresponding PPK ID.
2. The initiator-side key source synchronizes the PPK to the responder-side key source using an out-of-band mechanism that is specific to the type of key source. The IKEv2 initiator communicates the PPK ID to the IKEv2 responder over IKEv2 using the RFC 8784 extensions.
3. The IKEv2 responder requests from its key source, the PPK corresponding to the PPK ID received from the IKEv2 initiator. The key source replies with the PPK corresponding to the PPK ID.
4. The IKEv2 initiator and responder mix the PPK in the key derivation, as specified in RFC 8784. The resulting IKEv2 and OTNsec session keys are quantum-safe.

Configure Dynamic PPK in IKEv2

Cisco Secure Key Import Protocol (SKIP) is a protocol that allows an encryption device to securely import keys from an external PPK source. The externally imported PPKs are known as dynamic PPKs. To use SKIP, the encryption devices must implement the SKIP client, and the PPK source must implement the SKIP server. SKIP allows the use of QKD devices or Cisco Session Key Service (SKS) servers as the source of PPKs.

Configuring Dynamic PPK using SKS SKIP

Use the following commands to configure the dynamic PPK for one or more peers or groups of peers, in the IKEv2 keyring.

configure terminal

keyring dynamic

```

peer name
ppk dynamic sks-profile-name [required]
pre-shared-key key-string
address {ipv4-address mask}
ikev2 profile name
match identity remote address {ipv4-address mask}
keyring ppk keyring-name
keyring keyring-name
sks profile profile-name type remote
kme server ipv4 ip-address port port-number
exit
exit

```

Example :

```

RP/0/RP0/CPU0:ios#configure terminal
RP/0/RP0/CPU0:ios(config)#keyring dynamic
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#peer peer1
RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#ppk dynamic qkd required
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#pre-shared-key cisco123!cisco123
RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#address 10.0.0.1 255.0.0.0
RP/0/1/CPU0:ios(config)#ikev2 profile test
RP/0/1/CPU0:ios(config-ikev2-profile-test)#keyring dynamic
RP/0/1/CPU0:ios(config-ikev2-profile-test)#keyring ppk dynamic
RP/0/1/CPU0:ios(config-ikev2-profile-name)#match address 10.0.0.1 255.255.255.0

RP/0/1/CPU0:ios(config)#sks profile qkd type remote
RP/0/1/CPU0:ios(config-sks-profile)#kme server ipv4 192.0.2.34 port 10001
RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#exit
RP/0/RP0/CPU0:ios(config)#exit

```

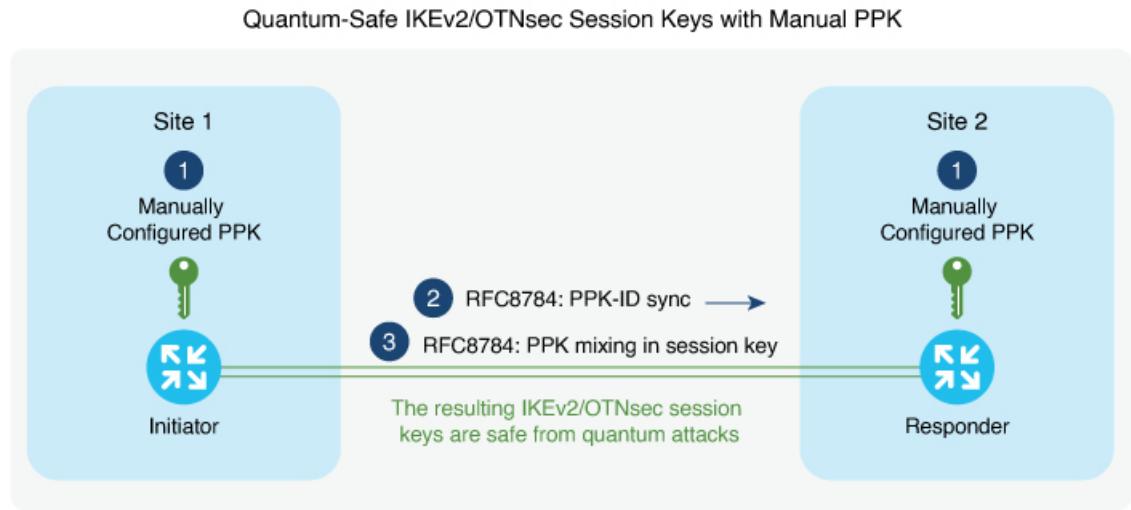
Manual Postquantum Preshared Keys

You can also use another easier way known as manual PPKs. When using the manual PPKs, you can provision the same PPKs on both the IKEv2 and OTNsec initiator and responder by manually configuring the PPKs on both sides.

Ensure that a manual PPK is of sufficient size, entropy, and is frequently rotated by the administrator.

In the following figure, you can see the session keys of quantum-safe IKEv2 and OTNsec, which are obtained through a manual PPK.

Figure 3: Quantum-Safe IKEv2 and OTNsec Session Keys with Manual PPK



Configure Manual PPK in IKEv2

Use the following commands to configure the manual PPK for one or more peers or groups of peers, in the IKEv2 keyring.



Note From Release 24.3.1, Type 6 passwords are supported on the preshared key. See [Enable Type 6 Password, on page 8](#) to enable the Type 6 method, which provides enhanced protection for the PPK and preshared key.

configure terminal

keyring *keyring-name*

peer *name*

ppk manual id *ppk-id* **key** [**clear** | **password** | **password6**] *password* [**required**]

pre-shared-key *key-string*

address {*ipv4-address mask* }

ikev2 profile *name*

match identity remote address {*ipv4-address mask*}

keyring ppk *keyring-name*

keyring *keyring-name*

exit

exit

Example :

```
RP/0/RP0/CPU0:ios#configure terminal
RP/0/RP0/CPU0:ios(config)#keyring manual
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#peer peer1
```

```

RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#ppk manual id cisco123 key password
060506324F41584B56 required
RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#pre-shared-key cisco123cisco123
RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#address 10.0.0.1 255.0.0.0
RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#exit
RP/0/RP0/CPU0:ios(config)#exit

RP/0/1/CPU0:ios(config)#ikev2 profile test
RP/0/1/CPU0:ios(config-ikev2-profile-test)#keyring manual
RP/0/1/CPU0:ios(config-ikev2-profile-test)#keyring ppk manual
RP/0/1/CPU0:ios(config-ikev2-profile-test)#match address 10.0.0.1 255.255.255.0
RP/0/RP0/CPU0:ios(config-ikev2-profile-test)#exit
RP/0/RP0/CPU0:ios(config)#exit

```

These are the examples where the PPK and the preshared key are configured with the `clear` keyword, after enabling the Type 6 method:

```

RP/0/RP0/CPU0:ios#configure terminal
RP/0/RP0/CPU0:ios(config)#keyring type6_psk
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#peer peer1
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#pre-shared-key clear cisco123
RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#address 10.0.0.1 255.0.0.0

RP/0/RP0/CPU0:ios#configure terminal
RP/0/RP0/CPU0:ios(config)#keyring type6_ppk
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#peer peer1
RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#ppk manual id 123 key clear cisco123 required
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#pre-shared-key

```

These are the examples where the PPK and preshared key are configured with the `password6` keyword, after enabling the Type 6 method:

```

RP/0/RP0/CPU0:ios#configure terminal
RP/0/RP0/CPU0:ios(config)#keyring type6_psk
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#peer peer1
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#pre-shared-key
password6
525548665b4e534660504c54645d63526668604945635a6452604a5f644d605a5c4461644d4e444e6566414142
RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#address 10.0.0.1 255.0.0.0

RP/0/RP0/CPU0:ios#configure terminal
RP/0/RP0/CPU0:ios(config)#keyring type6_ppk
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#peer peer1
RP/0/RP0/CPU0:ios(config-ikev2-keyring-peer)#ppk manual id 123 key password6
426447414e60494d48655d434f4749525d69484f434d445850675258544d56444a5d4d5b664b4c55624e414142
required
RP/0/RP0/CPU0:ios(config-ikev2-keyring)#pre-shared-key

```



Note When using the `password6` keyword for PPK and preshared key, you must use an encrypted key. You can use the encrypted form of the `clear` PPK and preshared key that you previously configured (retrieve the encrypted form using the **show running-config keyring** command), as long as the primary key used to enable the type 6 password remains the same.

Type 6 Password Support for Preshared Keys in IKEv2 Encryption

Table 2: Feature History

Feature Name	Release Information	Description
Type 6 Password Support for Keyring Configuration used in IKEv2	Cisco IOS XR Release 24.3.1	The keyring configuration for IKEv2 encryption now supports the Type 6 password encryption method. This method uses AES256-GCM encryption and a user-configured primary key to encrypt the preshared key and Postquantum Preshared Keys (PPK). This Type 6 encryption enhances security by storing sensitive information, such as the preshared key, in an encoded format on the device and makes it difficult to decipher. You can enable this feature using the following commands: • key config-key password-encryption • password6 aes encryption You can verify the status of the encryption using the following command: • show type6 server

From Release 24.3.1, the Type 6 password is supported on the PPK and preshared key used in the keyring configuration for the IKEv2 encryption. The Type 6 method uses a user-configured primary key to encrypt sensitive configurations, such as the preshared key, using the Advanced Encryption Standard (AES256-GCM) method. This primary key is stored in Trusted-Anchor-Module (TAM) and is never displayed in the configuration, ensuring that the Type 6 method is more secure and that the encrypted configuration is highly protected.

Enable Type 6 Password

You can enable the Type 6 password by configuring AES and the primary key.

Procedure

Step 1 Enter the global configuration mode.

Example:

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

Step 2 Enable the AES encryption and save the changes.

Example:

```
RP/0/RP0/CPU0:ios(config)#password6 encryption aes
RP/0/RP0/CPU0:ios(config)#commit
RP/0/RP0/CPU0:ios(config)#end
```

Step 3 Create the primary key.

Example:

```
P/0/0/CPU0:ios#key config-key password-encryption
Thu Jul 11 09:40:47.396 UTC
New password Requirements: Min-length 6, Max-length 64
Enter new key :
Enter confirm key :
Master key operation is started in background
```

Step 4 If you want to update the primary key, use the same command **key config-key password-encryption** again. When it prompts for an old key, input the old key and then enter the new key that you want to configure.

```
RP/0/RP0/CPU0:ios#key config-key password-encryption
Thu Jul 11 09:40:47.396 UTC
New password Requirements: Min-length 6, Max-length 64
Enter old key :
Enter new key :
Enter confirm key :
Master key operation is started in background
```

If you forget the old key while updating the primary key, use the command **key config-key password-encryption delete** to delete the old key and create a new one. Make sure that you disable **password6 encryption aes** before deleting the primary key.

Example:

```
RP/0/RP0/CPU0:ios#key config-key password-encryption delete
Thu Jul 11 09:42:39.612 UTC
Disable 'password6 encryption aes' before deleting master key, Masterkey delete Failed
RP/0/RP0/CPU0:ios#con
Thu Jul 11 10:01:47.056 UTC
RP/0/RP0/CPU0:ios(config)#no password6 encryption aes
RP/0/RP0/CPU0:ios(config)#commit
Thu Jul 11 10:01:57.755 UTC
RP/0/RP0/CPU0:ios(config)#end
RP/0/RP0/CPU0:ios#key config-key password-encryption delete
Thu Jul 11 10:02:02.238 UTC
WARNING: All type 6 encrypted keys will become unusable
Continue with master key deletion ? [yes/no]:yes
Master key operation is started in background
```

Enable the AES encryption and configure the primary key again. See [Step 2, on page 9](#) and [Step 3, on page 9](#)

Step 5 Verify the status of the Type 6 password encryption information.

Example:

```
RP/0/RP0/CPU0:ios#show type6 server
Thu Jul 11 09:43:36.103 UTC
Server detail information:
=====
AES config State      : Enabled
Masterkey config State : Enabled
```

```
Type6 feature State      :      Enabled
Master key Inprogress   :      No
Masterkey Last updated/deleted : Thu Jul 11 09:40:57 2024
```

When the key is created, it is stored internally; not as part of the NCS 1004 device configuration. The device does not display the primary key as part of the running configuration. So, you cannot see or access the primary key when you connect to the device.

The configured primary key and AES encryption encrypt the preshared key and PPK. These are the examples where the **show running-config keyring** command displays the preshared key and PPK configured during the [Manual PPK](#) configuration in the encrypted format.

```
RP/0/RP0/CPU0:ios#show running-config keyring
Thu Jul 11 09:42:43.085 UTC
keyring keyring type6_psk
peer link-1
  pre-shared-key password6 62415744534e564365625c544e5b68594d4158515d4f6768436845556747414142
  address 1.1.1.2 255.255.255.0

RP/0/RP0/CPU0:ios#show running-config keyring
Thu Jul 11 09:42:43.085 UTC
keyring type6_ppk
peer 1
  ppk manual id 123 key password6
  4d49525f5848444b535b625243584764636952644e64414952415641554655635e5f4c56424d46455c65414142 required
  !
```

Note

Use the nonvolatile generation (NVGEN) command, if you want to mask completely the preshared key in the show command output.

Example:

```
RP/0/0/CPU0:ios(config)#nvgen-default-sanitize passwords
RP/0/0/CPU0:ios(config)#commit

RP/0/0/CPU0:ios(config)#show running-config keyring
Thu Jul 11 09:45:37.193 UTC
keyring keyring_all_in_one
peer link-1
  pre-shared-key password6 <removed>
  address 1.1.1.2 255.255.255.0
  !
!
```

Verify the PPK Configuration

This section describes the commands to verify the configured PPK details.

View the Current IKEv2 Security Associations

Use the **show ikev2 sa detail** command to display information about the current IKEv2 security associations. The `Quantum resistance` parameter in the output of the command indicates that manual PPK-based quantum-safe encryption is enabled.



Note Both manual and dynamic PPK options can be used for viewing IKEv2 details.

The following is a sample output from the **show ikev2 sa detail** command:

```
RP/0/1/CPU0:ios#show ikev2 sa detail
```

```

IKE SA ID                               : 866
-----
Local                                   : 192.0.2.34/500
Remote                                 : 192.0.2.40/500
Status (Description)                   : READY (Negotiation done)
Role                                   : Initiator
Fvrf                                   : Default
Encryption/Keysize                     : AES-CBC/256
PRF/Hash/DH Group                      : SHA512/SHA512/19
Authentication (Sign/Verify)           : PSK/PSK
Life/Active Time (sec)                 : 86400/21
Session ID                             : 5
Local SPI                              : C18D2946B0C4259C
Remote SPI                             : 5D1BD398AEB3A1E1
Local ID                               : 192.0.2.34
Remote ID                              : 192.0.2.40
Quantum resistance                     : Enabled with manual PPK

```

View IKEv2 Session Statistics

Use the **show ikev2 statistics** command to display the statistics and counters related to IKEv2 sessions.

The following is a sample output from the **show ikev2 statistics** command:

```
RP/0/1/CPU0:ios#show ikev2 statistics
```

```
Thu Jun  8 13:30:06.360 IST
```

```

.....

NO_NAT                                :          2          0          0          0

PPK COUNTERS
=====

PPK ERRORS
-----
PPK_ID_MISMATCH                       :          0
PPK_RETRIEVE_FAIL                     :          0
PPK_AUTH_FAIL                         :          0

```

View IKEv2 Session Summary

Use the **show ikev2 summary** command to display the IKEv2 session summary of NCS 1014.

The following is a sample output from the **show ikev2 summary** command:

```
RP/0/1/CPU0:ios#show ikev2 summary
```

```
Thu Jun  8 12:54:30.969 IST
```

```
IKEv2 SA Summary
```

```
-----
```

```

Total SA (Active/Negotiating)      : 2 (2/0)
Total Outgoing SA (Active/Negotiating) : 2 (2/0)
Total Incoming SA (Active/Negotiating) : 0 (0/0)
Total QR SA (Dynamic/Manual)       : 2 (1/1)

```

View IKEv2 Profile Details

Use the **show ikev2 profile** command to display all the IKEv2 profile details.

The following is a sample output from the **show ikev2 profile** command:

```

RP/0/1/CPU0:ios#show ikev2 profile
Tue Jun  6 18:00:20.277 IST

```

```

Profile Name                               : p4
=====
Keyring                                    : k4
Fvrf                                       : Default
Lifetime(Sec)                             : 86400
DPD Interval(Sec)                         : 4
DPD Retry Interval(Sec)                   : 2
Match ANY                                  : NO
Total Match remote peers                   : 1
  Addr/Prefix                             : 198.51.100.19/255.255.255.0
Number of Trustpoints                     : 0
Local auth method                         : PSK
Number of remote auth methods              : 1
  Auth Method                             : PSK
PPK Keyring                               : Not Configured

Profile Name                               : ppk_d
=====
Keyring                                    : Not Configured
Fvrf                                       : Default
Lifetime(Sec)                             : 86400
DPD Interval(Sec)                         : 4
DPD Retry Interval(Sec)                   : 2
Match ANY                                  : NO
Total Match remote peers                   : 0
Number of Trustpoints                     : 0
Local auth method                         : NULL
Number of remote auth methods              : 0
PPK Keyring                               : ppk_d

Profile Name                               : ppk_m
=====
Keyring                                    : Not Configured
Fvrf                                       : Default
Lifetime(Sec)                             : 86400
DPD Interval(Sec)                         : 4
DPD Retry Interval(Sec)                   : 2
Match ANY                                  : NO
Total Match remote peers                   : 0
Number of Trustpoints                     : 0
Local auth method                         : NULL
Number of remote auth methods              : 0
PPK Keyring                               : ppk_m

```

View Keyring Details

Use the **show keyring** command to display the configured keyring details on NCS 1014.

The following is a sample output from the **show keyring** command:

```
RP/0/1/CPU0:ios#show keyring
Tue Jun  6 18:00:28.272 IST
```

```
Keyring Name                               : k4
=====
Total Peers                               : 1
-----
  Peer Name                               : init
  IP Address                               : 198.51.100.19
  Subnet Mask                              : 255.255.255.0
  Local PSK                                : Configured
  Remote PSK                               : Configured
  PPK Mode                                 : Not Configured
  PPK Mandatory                            : Not Configured

Keyring Name                               : ppk_m
=====
Total Peers                               : 1
-----
  Peer Name                               : init
  IP Address                               : Not Configured
  Subnet Mask                              : Not Configured
  Local PSK                                : Not Configured
  Remote PSK                               : Not Configured
  PPK Mode                                 : Manual
  PPK Mandatory                            : No

Keyring Name                               : ppk_m_req
=====
Total Peers                               : 1
-----
  Peer Name                               : init
  IP Address                               : Not Configured
  Subnet Mask                              : Not Configured
  Local PSK                                : Not Configured
  Remote PSK                               : Not Configured
  PPK Mode                                 : Manual
  PPK Mandatory                            : Yes

Keyring Name                               : ppk_d
=====
Total Peers                               : 1
-----
  Peer Name                               : init
  IP Address                               : Not Configured
  Subnet Mask                              : Not Configured
  Local PSK                                : Not Configured
  Remote PSK                               : Not Configured
  PPK Mode                                 : Dynamic
  PPK Mandatory                            : No

Keyring Name                               : ppk_d_req
=====
Total Peers                               : 1
-----
  Peer Name                               : init
  IP Address                               : Not Configured
  Subnet Mask                              : Not Configured
  Local PSK                                : Not Configured
```

Remote PSK	: Not Configured
PPK Mode	: Dynamic
PPK Mandatory	: Yes

View IKEv2 Session Detail

Use the **show ikev2 session detail** command to display information about the current IKEv2 session.

The following is a sample output from the **show ikev2 session detail** command:

```
RP/0/1/CPU0:ios#show ikev2 session detail
Fri Feb  2 11:21:09.131 IST
Session ID                               : 3
=====
Status                                   : UP-ACTIVE
IKE Count                               : 1
Child Count                             : 1
IKE SA ID                               : 11625
-----
Local                                   : 192.0.2.3/500
Remote                                  : 192.0.2.1/500
Status (Description)                    : READY (Negotiation done)
Role                                    : Initiator
Fvrf                                    : Default
Encryption/Keysize                      : AES-CBC/256
PRF/Hash/DH Group                       : SHA512/SHA512/19
Authentication (Sign/Verify)            : PSK/PSK
Life/Active Time (sec)                  : 200/115
Session ID                              : 3
Local SPI                              : E8F0716FF44EA1C3
Remote SPI                              : B1046E13B805178E
Local ID                                : 192.0.2.3
Remote ID                               : 192.0.2.1
Quantum resistance                       : Enabled with manual PPK
-----
Child SA
-----
Local Selector                          : 0.0.0.0/0 - 255.255.255.255/65535
Remote Selector                         : 0.0.0.0/0 - 255.255.255.255/65535
ESP SPI IN/OUT                          : 0xf5e2a1c2 / 0x12bb94fd
Encryption                              : AES-CBC
Keysize                                 : 256
ESP HMAC                                : SHA384
```