

# Configure DMVPN Phase 3 Using IKEv2 with Certificate Authentication

## Contents

---

### [Introduction](#)

### [Prerequisites](#)

[Requirements](#)

[Components Used](#)

### [Background Information](#)

### [Configure](#)

[Network Diagram](#)

[Configurations](#)

[Prepare Certificate Infrastructure](#)

[Crypto IKEv2 and IPSec Configuration](#)

[Tunnel Configuration](#)

### [Verify](#)

### [Troubleshoot](#)

---

## Introduction

This document describes information on how to configure Dynamic Multipoint VPN (DMVPN) phase 3 with certificate authentication using IKEv2.

## Prerequisites

### Requirements

Cisco recommends having knowledge of the topics:

- Basic knowledge of DMVPN.
- Basic knowledge of EIGRP.
- Basic Knowledge of Public Key Infrastructure (PKI).

### Components Used

The information in this document is based on this software version:

- Cisco C8000v (VXE) running Cisco IOS® Version 17.3.8a.

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

## Background Information

Dynamic Multipoint VPN (DMVPN) Phase 3 introduces direct Spoke to Spoke connectivity, enabling VPN networks to operate more efficiently by bypassing the Hub for most traffic paths. This design minimizes latency and optimizes resource utilization. The use of Next Hop Resolution Protocol (NHRP) allows spokes to dynamically identify each other and create direct tunnels, supporting large and complex network topologies.

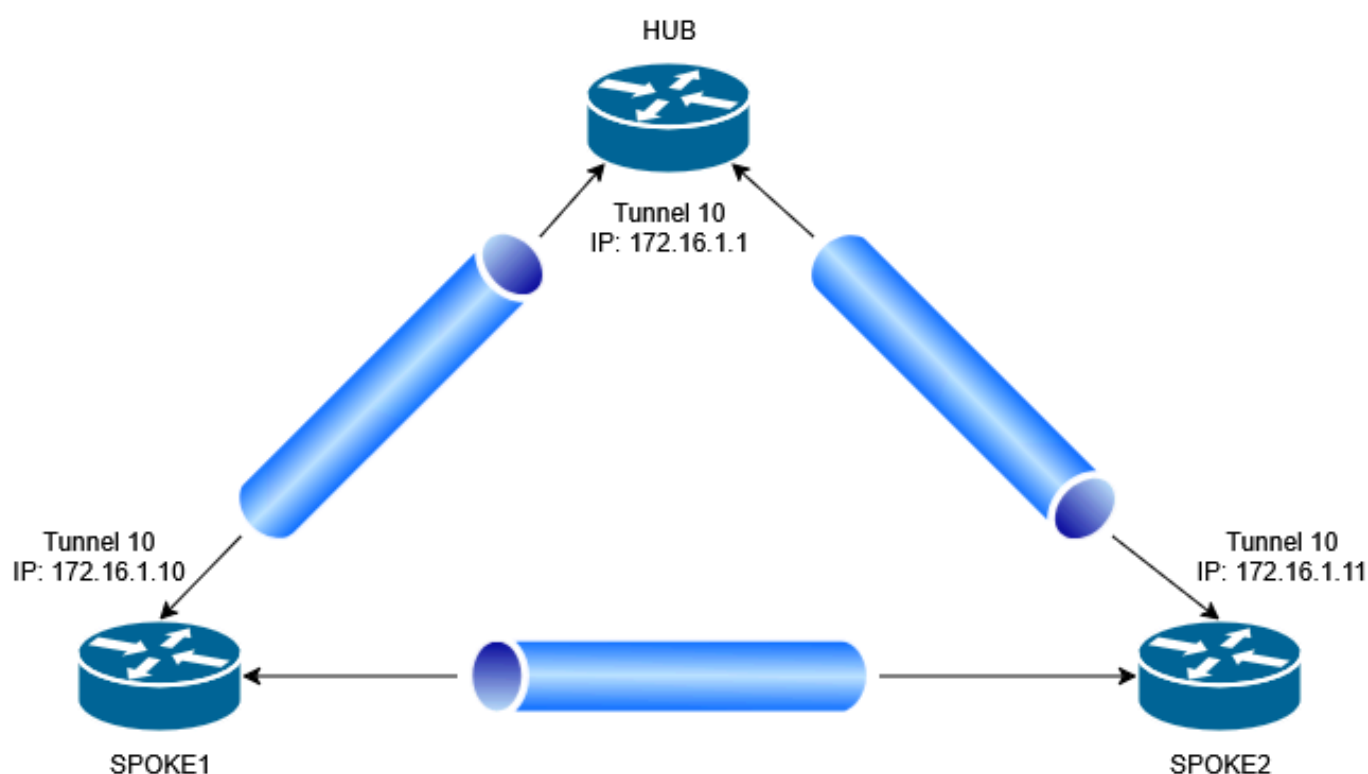
Internet Key Exchange version 2 (IKEv2) provides the underlying mechanism for establishing secure tunnels in this environment. Compared to earlier protocols, IKEv2 offers advanced security measures, quicker rekeying processes, and enhanced support for both mobility and multiple connections. Its integration with DMVPN Phase 3 ensures that tunnel setup and key management are handled securely and effectively. To further strengthen network security, IKEv2 supports digital certificate authentication. This approach enables devices to verify identities between each others using certificates, which simplifies management and reduces the risks associated with shared secrets. Certificate based trust is especially beneficial in expansive deployments where managing individual keys would be challenging.

Altogether, DMVPN Phase 3, IKEv2, and certificate authentication deliver a robust VPN framework. This solution addresses the requirements of modern enterprises by ensuring flexible connectivity, strong protection of data, and streamlined operations.

## Configure

This section provides step by step instructions to configure DMVPN Phase 3 with IKEv2 using certificate based authentication. Complete these steps to enable secure and scalable VPN connectivity between Hub and Spoke routers.

### Network Diagram



## Configurations

### Prepare Certificate Infrastructure

Ensure that all devices (Hubs and Spokes) have the necessary digital certificates installed. These certificates must be issued by a trusted CA and properly enrolled on each device to enable secure IKEv2 certificate authentication.

To enroll a certificate on Hub and Spoke routers, complete these steps:

1. Configure a **trustpoint** with the required information using the command **crypto pki trustpoint <Trustpoint Name>**.

```
<#root>
```

```
Hub(config)#
```

```
crypto pki trustpoint myCertificate
```

```
Hub(ca-trustpoint)# enrollment terminal
```

```
Hub(ca-trustpoint)# ip-address 10.10.1.2
```

```
Hub(ca-trustpoint)# subject-name cn=Hub, o=cisco
```

```
Hub(ca-trustpoint)# revocation-check none
```

2. Authenticate the **trustpoint** using the command **crypto pki authenticate <Trustpoint Name>**.

```
<#root>
```


```
Hub(config)#
```

```
crypto pki authenticate myCertificate
```

Enter the base 64 encoded CA certificate.

End with a blank line or the word "quit" on a line by itself

---

 **Note:** After issuing the command **crypto pki authenticate** you must paste the certificate from Certificate Authority (CA) that is used to sign the device certificates. This step is essential to establish trust between the device and the CA before proceeding with certificate enrollment on both Hub and Spoke routers.

---

3. Generate the **private key** and **Certificate Signing Request (CSR)** by using the command **crypto pki enroll <Trustpoint Name>**.

```
<#root>
```

```
Hub(config)#
```

```
crypto pki enroll myCertificate
```

```
% Start certificate enrollment ..
```

```
% The subject name in the certificate will include: cn=Hub, o=cisco
```

```
% The subject name in the certificate will include: Hub
```

```
% Include the router serial number in the subject name? [yes/no]: n
```

```
% The IP address in the certificate is 10.10.1.2
```

Display Certificate Request to terminal? [yes/no]: yes  
Certificate Request follows:

```
MIICsDCCAZgCAQAwSjEOMAwGA1UEChMFY2l2Y28xDDAKBgNVBAMTA0hVQjEqMBAG
CSqGSIb3DQEJAhYDSFVCMBYGCsGSIb3DQEJCBMjMTAuMTAuMS4yMIIBIjANBgkq
hkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAO/M40+ivsqJhpF0PRUxdCGSUVgLUHzQ
cwnuMtSbfdn5fMKIj7w06Qa7Gvx2rjrdoyxH9JgXjTEMzMv6HP9/EuN2o+qKzR/+
CNzMUDJobb01BNbe0WKL4IAQjbvNT0yA5iuUzHZCgMrCFG3oU7v+a2tMiSZihvdu
+m2JSDNXn5cXyewQbQsEaELA00yosi2t6BQyzM3FRU23dCwnFVwY1VAADC7CrNh3
o44SifYw5HtWq1tU1cLTY4sjNf6XJQxjmHPudbUp164RDFUSo37Zjvjt7S800oLU
+XUBrE3aRDlwJ+Ug2D031ZWzfc+rBZ1BsKW1YFB1Lk3mL9RA1nf3eQIDAQABoCEw
HwYJKoZIhvcNAQkOMRIwEDA0BgNVHQ8BAf8EBAMCBaAwDQYJKoZIhvcNAQEFBQAD
ggEBAEKUQUURWZ+YeCx9T7kuzIaDwJ53vMqq6rITDJcNF9FJ4IgJ7PsxF0cWxm7MM
030i1yq1K/4X7Mb5Iz6CjtdyXVqakgcEPY7W9No03Xo8Nxb4pFfe19E02XuJ8fxm
GTqi7UAw8Zs1zJ2jrS7bXasVMb5j39cqQkrXfNIAwF1Sw6IA3oKfTe1q8/iCJu
TEJfOD8Si2PWziuxJVS4Adjg5GxbJpd/tDKrKUuvqD2z4HD3M40oGVvoBWQ0tjhB
4gx1q2D209K0nMCvVZr0fp/PFd6+cYc57E73ZPVSGQpHIiWcYtuRKdKArN6vRcP
iiugceU2F3L14CI7wXMYqCxCQOGU=
```

---End - This line not part of the certificate request---

Redisplay enrollment request? [yes/no]:



**Note:** The private key used during this process is the default private key generated by the router. However, the use of custom private keys is also supported, if required.

4. After generating the **CSR**, send it to the Certificate Authority (CA) to be signed.

5. Once the certificate is signed, use the command **crypto pki import <Trustpoint Name> certificate** to import the signed certificate associated with the trustpoint created.

```
<#root>
```

```
Hub(config)#
```

```
crypto pki import myCertificate certificate
```

% You must authenticate the Certificate Authority before  
you can import the router's certificate.

6. Paste the **certificate** signed by the CA in PEM format.

## Crypto IKEv2 and IPSec Configuration

The configuration for Crypto IKEv2 and IPSec can be the same on both the spokes and the Hub. This is because elements such as the proposals and the ciphers used must always match across all devices to ensure the tunnel can be successfully established. This consistency guarantees interoperability and secure communication within the DMVPN Phase 3 environment.

1. Configure an **IKEv2 proposal**.

```
crypto ikev2 proposal ikev2
encryption aes-cbc-256
integrity sha256
group 14
```

## 2. Configure an **IKEv2 profile**.

<#root>

```
crypto ikev2 profile ikev2Profile
match identity remote address 0.0.0.0
identity local address 10.10.1.2
```

```
authentication remote rsa-sig
```

```
authentication local rsa-sig
```

```
pki trustpoint
```

```
myCertificate
```



**Note:** Here is where PKI certificate authentication is defined and the trustpoint that is used for authentication.

---

## 3. Configure an **IPSec profile** and a **transform set**.

```
crypto ipsec transform-set ipsec esp-aes 256 esp-sha256-hmac
mode tunnel
crypto ipsec profile ipsec
set transform-set ipsec
set ikev2-profile ikev2Profile
```

## Tunnel Configuration

This section covers the configuration of tunnels for both the Hub and the Spokes, specifically focusing on Phase 3 of the DMVPN setup.

### 1. Hub tunnel configuration.

```
interface Tunnel10
ip address 172.16.1.1 255.255.255.0
no ip redirects
no ip split-horizon eigrp 10
ip nhrp authentication cisco123
ip nhrp network-id 10
ip nhrp redirect
tunnel source GigabitEthernet1
```

```
tunnel mode gre multipoint
tunnel protection ipsec profile ipsec
end
```

## 2. Spoke1 tunnel configuration.

```
interface Tunnel10
ip address 172.16.1.10 255.255.255.0
no ip redirects
ip nhrp authentication cisco123
ip nhrp map 172.16.1.1 10.10.1.2
ip nhrp map multicast 10.10.1.2
ip nhrp network-id 10
ip nhrp nhs 172.16.1.1
tunnel source GigabitEthernet2
tunnel mode gre multipoint
tunnel protection ipsec profile ipsec
end
```

## 3. Spoke2 tunnel configuration.

```
interface Tunnel10
ip address 172.16.1.11 255.255.255.0
no ip redirects
ip nhrp authentication cisco123
ip nhrp map 172.16.1.1 10.10.1.2
ip nhrp map multicast 10.10.1.2
ip nhrp network-id 10
ip nhrp nhs 172.16.1.1
tunnel source GigabitEthernet3
tunnel mode gre multipoint
tunnel protection ipsec profile ipsec
end
```

# Verify

To confirm that the DMVPN Phase 3 network is functioning correctly, use these commands:

- **show dmvpn interface <Tunnel Name>**
- **show crypto ikev2 sa**
- **show crypto ipsec sa peer <peer IP>**

With the **show dmvpn interface <Tunnel Name>** command, you can see the active sessions between the Hub and the spokes. From the perspective of Spoke1, the output can reflect these established connections.

<#root>

SPOKE1#

```
show dmvpn interface tunnel10
```

Legend: Attrb --> S - Static, D - Dynamic, I - Incomplete  
N - NATed, L - Local, X - No Socket  
T1 - Route Installed, T2 - Nexthop-override, B - BGP  
C - CTS Capable, I2 - Temporary  
# Ent --> Number of NHRP entries with same NBMA peer  
NHS Status: E --> Expecting Replies, R --> Responding, W --> Waiting  
UpDn Time --> Up or Down Time for a Tunnel  
=====

Interface: Tunnel10, IPv4 NHRP Details  
Type:Spoke, NHRP Peers:2,

# Ent Peer NBMA Addr Peer Tunnel Add

State

UpDn	Tm	Attrb	
1	10.10.1.2		172.16.1.1

UP

	1w6d	S	
1	10.10.3.2		172.16.1.11

UP

00:00:04	D
----------	---

The **show crypto ikev2 sa** command displays the IKEv2 tunnels formed between the spokes and the Hub, confirming successful Phase 1 negotiations.

<#root>

SPOKE1#

show crypto ikev2 sa

IPv4 Crypto IKEv2 SA

Tunnel-id	Local	Remote	fvr/ivrf
-----------	-------	--------	----------

Status

1	10.10.2.2/500	10.10.3.2/500	none/none
---	---------------	---------------	-----------

READY

Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:19, Auth sign: RSA, Auth verify:

RSA

Life/Active Time: 86400/184 sec

Tunnel-id	Local	Remote	fvr/ivrf
-----------	-------	--------	----------

## Status

2	10.10.2.2/500	10.10.1.2/500	none/none
---	---------------	---------------	-----------

## READY

Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:19, Auth sign: RSA, Auth verify:

## RSA

Life/Active Time: 86400/37495 sec

IPv6 Crypto IKEv2 SA

Using the **show crypto ipsec sa peer <peer IP>** command, you can verify the IPSec tunnels established between the spokes and the Hub, ensuring secure data transport within the DMVPN network.

<#root>

SPOKE1#show

crypto ipsec sa peer 10.10.3.2

interface: Tunnel10

Crypto map tag: Tunnel10-head-0, local addr 10.10.2.2

protected vrf: (none)

local ident (addr/mask/prot/port): (10.10.2.2/255.255.255.255/47/0)

remote ident (addr/mask/prot/port): (10.10.3.2/255.255.255.255/47/0)

current\_peer 10.10.3.2 port 500

PERMIT, flags={origin\_is\_acl,}

#pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4

#pkts decaps: 5, #pkts decrypt: 5, #pkts verify: 5

#pkts compressed: 0, #pkts decompressed: 0

#pkts not compressed: 0, #pkts compr. failed: 0

#pkts not decompressed: 0, #pkts decompress failed: 0

#send errors 0, #recv errors 0

local crypto endpt.: 10.10.2.2, remote crypto endpt.: 10.10.3.2

plaintext mtu 1438, path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet2

current outbound spi: 0xF341E02E(4081180718)

PFS (Y/N): N, DH group: none

inbound esp sas:

spi: 0x8ED55E26(2396347942)

transform: esp-256-aes esp-sha256-hmac ,

in use settings ={Tunnel, }

conn id: 2701, flow\_id: CSR:701, sibling\_flags FFFFFFFF80000048, crypto map: Tunnel10-head-0

sa timing: remaining key lifetime (k/sec): (4607999/3188)

IV size: 16 bytes

replay detection support: Y

Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcp sas:



outbound esp sas:  
spi: 0xF341E02E(4081180718)  
transform: esp-256-aes esp-sha256-hmac ,  
in use settings ={Tunnel, }  
conn id: 2702, flow\_id: CSR:702, sibling\_flags FFFFFFFF80000048, crypto map: Tunnel10-head-0  
sa timing: remaining key lifetime (k/sec): (4607999/3188)  
IV size: 16 bytes  
replay detection support: Y  
Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:

## Troubleshoot

For troubleshooting, you can use these commands:

- **debug dmvpn condition peer [nbma/tunnelIP]**, enables conditional debugging for DMVPN sessions specific to an NBMA or tunnel IP address from a peer, helping isolate issues related to that peer.
- **debug dmvpn all all**, enables comprehensive debugging for all aspects of DMVPN, including NHRP, crypto IKE, IPsec, tunnel protection, and crypto sockets. It is recommended to use this command with a conditional filter to avoid overwhelming the router with excessive debug information.
- **show dmvpn**, Displays the current DMVPN status, including tunnel interfaces, NHRP mappings, and peer information.
- **show crypto ikev2 sa**, Shows the status of IKEv2 Security Associations, useful for verifying Phase 1 VPN negotiations.
- **show crypto ipsec sa**, Displays IPsec Security Associations, showing Phase 2 tunnel status and traffic statistics.