Sharing IPsec with Tunnel Protection

The Sharing IPsec with Tunnel Protection feature allows sharing an IPsec security association database (SADB) between two or more generic routing encapsulation (GRE) tunnel interfaces when tunnel protection is used. Shared tunnel interfaces have a single underlying cryptographic SADB, cryptographic map, and IPsec profile in the Dynamic Multipoint Virtual Private Network (DMVPN) configuration.

The Sharing IPsec with Tunnel Protection feature is required in some DMVPN configurations. If IPsec SA sessions are not shared within the same IPsec SADB, an IPsec SA may be associated with the wrong IPsec SADB and therefore with the wrong tunnel interface, thereby causing duplicate IPsec security associations (SAs) and tunnel interfaces to flap, which in turn results in network connectivity problems.

Note Security threats and the cryptographic technologies to help protect against such threats are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Restrictions for Sharing IPsec with Tunnel Protection

- If two or more generic route encapsulation (GRE) tunnel interfaces share the same tunnel source interface and one of the GRE tunnel interface is an multipoint generic route encapsulation (mGRE) tunnel interface, all tunnels with the same tunnel source must use different tunnel keys, the same IPsec profile name, and the **shared** keyword with the **tunnel protection** command.
- If there are multiple point-to-point GRE tunnel interfaces that share the same tunnel source interface with the same tunnel destination address, the GRE tunnels must use different tunnel keys, the same IPsec profile name, and the **shared** keyword in the **tunnel protection** command.
- Shared tunnel protection is not required and should not be used when several point-to-point GRE tunnels share the same tunnel source but have unique tunnel destination IP addresses.
- The tunnel source command on all tunnel interfaces that use shared tunnel protection must be configured using the interface type and number and not the IP address.

Note It is recommended that the **tunnel source** command be configured with an interface than an IP address on all GRE tunnels.

- Different IPsec profile names must be used for shared and unshared tunnels. For example, if "tunnel 1" is configured with the **tunnel source loopback1** command, and "tunnel 2" and "tunnel 3" are shared using the **tunnel source loopback2** command, use separate IPsec profiles, for example, define IPsec_profile_1 for tunnel 1 and IPsec_profile_2 for tunnels 2 and 3.
- Different IPsec profile must be used for each set of shared tunnels. For example, if tunnels 1 through 5 use tunnel source loopback1 and tunnels 6 through 10 use tunnel source loopback2, use IPsec_profile_1 for tunnels 1 through 5 and ipsec_profile_2 for tunnels 6 through 10.
- There are few exceptions to the above rules:
  - Several mGRE tunnels sharing the same tunnel source interface can be configured without the **shared** keyword in the **tunnel protection** command if they use different IPsec profiles with different IPsec transform sets. Different IPsec transform sets disambiguate tunnel setup in this case. Each mGRE tunnel interface must still be configured with a different tunnel key. This applies to several mGRE tunnels and point-to-point GRE tunnels sharing the same tunnel source. This method cannot be used if several point-to-point GRE tunnels share the same tunnel source interface and the same tunnel destination address.
  - Sometimes, it may be desirable not to share an IPsec session between two or more tunnel interfaces using the same tunnel source. For example, in a service provider environment, each DMVPN cloud can represent a different customer. It is desirable to lock the connections from a customer to a tunnel interface and not share or allow IPsec sessions from other customers. In such scenarios, Internet Security Association and Key Management Protocol (ISAKMP) profiles can be used to identify and bind customer connections to an ISAKMP profile and through that to an IPsec profile. This ISAKMP profile limits the IPsec profile to accept only those connections that match the corresponding ISAKMP profile. Separate ISAKMP and IPsec profiles can be obtained for each DMVPN cloud (tunnel interface) without sharing the same IPsec Security Association Database (SADB).

Note An exception is multiple ISAKMP sessions between same peers, which will not work. For example, in a dual hub dual DMVPN setup, the security associations (SAs) for the second tunnel interface between the hubs will not come up without sharing the SADB. Hence, the hubs cannot register to themselves on both mGRE tunnel interfaces without using the **shared** keyword in the IPsec profile.

- Shared tunnel protection is not supported for a IPsec virtual tunnel interface (VTI). If there are VTI tunnels sharing the same tunnel source with other GRE or mGRE tunnels that have shared tunnel protection, these VTI tunnels should be configured with different IPsec profiles without using the **shared** keyword.

Information About Sharing IPsec with Tunnel Protection

Single IPsec SA

In a dual-hub, dual-DMVPN topology, it is possible to have two or more generic route encapsulation (GRE) tunnel sessions (same tunnel source and destination, but different tunnel keys) between the same two endpoints. In this case, it is desirable to use a single IPsec SA to secure both GRE tunnel sessions. It is also not possible to decide under which tunnel interface an IPsec Quick Mode (QM) request must be processed and bound when two tunnel interfaces use the same tunnel source.

The **tunnel protection IPsec profile shared** command is used to create a single IPsec SADB for all the tunnel interfaces that use the same profile and tunnel source interface. This allows a single IPsec SA to be used for all GRE tunnels (same tunnel source and destination, but different tunnel keys) between the same two endpoints. It also makes IPsec QM processing unambiguous because there is one SADB to process the incoming IPsec QM request for all shared tunnel interfaces as opposed to multiple SADBs, one for each tunnel interface when the tunnel interface is not shared.

The SA of a QM proposal to a tunnel interface is processed by using the shared SADB and crypto map parameters. On the crypto-data plane, the decrypted and GRE decapsulated packets are demultiplexed to the appropriate tunnel interface by the GRE module using a local address, remote address, and optional tunnel key information.

Note The tunnel source, tunnel destination, and tunnel key (triplet) must be unique for all tunnel interfaces on a device. For a multipoint GRE interfaces where the tunnel destination is not configured, the pair (tunnel source and tunnel key) must be unique. Incoming GRE packets are also matched to point-to-point GRE tunnels first; if there is no match, they are matched to mGRE tunnels.

How to Share an IPsec Session Between Multiple Tunnels

Sharing an IPsec SADB Between Multiple Tunnel Interfaces in a DMVPN

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **tunnel source** {*ip-address* | *interface-type number*}
5. **tunnel protection IPsec profile** *name* **shared**
6. **end**
7. Repeat this task to configure additional spokes.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 5`	Configures a tunnel interface and enters interface configuration mode. • The *number* argument specifies the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces that you can create.
Step 4	**tunnel source** {*ip-address* | *interface-type number*} **Example:** `Device(config-if)# tunnel source Ethernet 0`	Sets the source IP address or source interface type and number for a tunnel interface. • When using the **tunnel protection IPsec profile shared** command, the tunnel source must specify an interface, not an IP address.
Step 5	**tunnel protection IPsec profile** *name* **shared** **Example:**	Associates a tunnel interface with an IPsec profile. • The *name* argument specifies the name of the IPsec profile; this value must match the *name* specified in the **crypto IPsec profile** *name* command.

	Command or Action	Purpose
	`Device(config-if)# tunnel protection IPsec profile vpnprof shared`	• The **shared** keyword allows IPsec sessions to be shared between multiple tunnel interfaces that are configured with the same tunnel source IP.
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.
Step 7	Repeat this task to configure additional spokes.	—

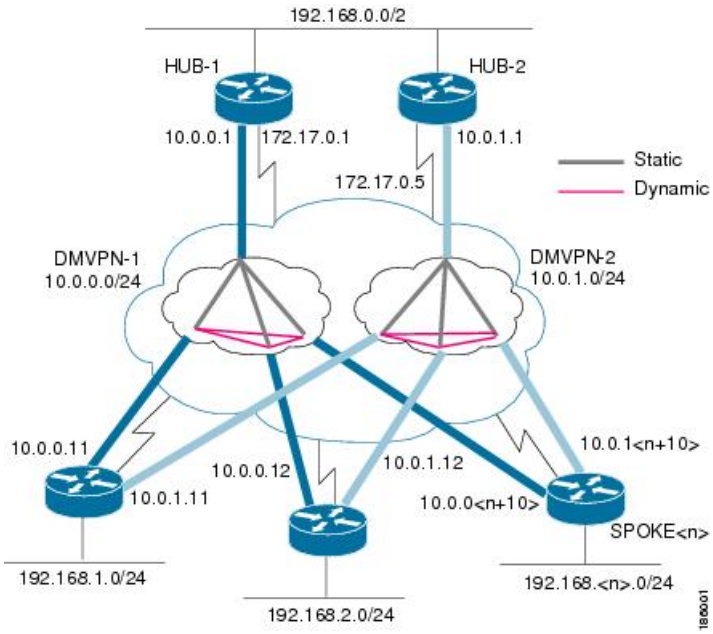
Configuration Examples for Sharing IPsec with Tunnel Protection

Example: Sharing IPsec Sessions Between Multiple Tunnels

The following example shows how to share IPsec sessions between multiple tunnels. This example uses the dual-hub router, dual-DMVPN topology as shown in the figure below and has the following attributes:

- Each hub device is configured with a single multipoint generic routing encapsulation (mGRE) tunnel interface.
- Each hub device is connected to one DMVPN subnet (blue cloud), and the spokes are connected to both DMVPN 1 and DMVPN 2.
- Each spoke device is configured with two mGRE tunnel interfaces.
- One mGRE tunnel interface belongs to DMVPN 1, and the other mGRE tunnel interface belongs to DMVPN 2.
- Each mGRE tunnel interface is configured with the same tunnel source IP address and uses shared tunnel protection between them.

Figure 1: Dual-Hub Router and Dual-DMVPN Topology



Hub 1 Configuration

The Hub 1 and Hub 2 configurations are similar, except that each hub belongs to a different DMVPN.

Hub 1 has the following DMVPN configuration:

- IP subnet: 10.0.0.0/24
- Next Hop Resolution Protocol (NHRP) network ID: 100000
- Tunnel key: 100000
- Dynamic routing protocol: Enhanced Interior Gateway Routing Protocol (EIGRP)

```
!
hostname Hub1
 !
 crypto isakmp policy 1
 encryption aes
 authentication pre-share
 group 14
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto IPsec transform-set trans2 esp-aes esp-sha-hmac
  mode transport
 !
 crypto IPsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.1 255.255.255.0
  ip mtu 1400
  no ip next-hop-self eigrp 1
```

```
  ip nhrp authentication test
  ip nhrp map multicast dynamic
  ip nhrp network-id 100000
  ip nhrp holdtime 600
no ip split-horizon eigrp 1
ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection IPsec profile vpnprof
 !
 interface Ethernet0
  ip address 172.16.0.1 255.255.255.252
 !
 interface Ethernet1
  ip address 192.168.0.1 255.255.255.0
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
  network 192.168.0.0 0.0.0.255
  no auto-summary
 !
```

Hub 2 Configuration

Hub 2 has the following DMVPN configuration:

- IP subnet: 10.0.1.0/24
- NHRP network ID: 100001
- Tunnel key: 100001
- Dynamic routing protocol: EIGRP

```
!
hostname Hub2
 !
 crypto isakmp policy 1
 encryption aes
 authentication pre-share
 group 14
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto IPsec transform-set trans2 esp-aes esp-sha-hmac
  mode transport
 !
 crypto IPsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.1.1 255.255.255.0
  ip mtu 1400
  no ip next-hop-self eigrp 1
  ip nhrp authentication test
  ip nhrp map multicast dynamic
  ip nhrp network-id 100001
  ip nhrp holdtime 600
no ip split-horizon eigrp 1
ip tcp adjust-mss 1360
```

```
 delay 1000
 tunnel source Ethernet 0
 tunnel mode gre multipoint
 tunnel key 100001
 tunnel protection IPsec profile vpnprof
!
interface Ethernet0
 ip address 172.16.0.5 255.255.255.252
!
interface Ethernet1
 ip address 192.168.0.2 255.255.255.0
!
router eigrp 1
 network 10.0.1.0 0.0.0.255
 network 192.168.0.0 0.0.0.255
 no auto-summary
!
```

Spoke 1 Configuration

Spoke 1 has the following DMVPN configuration:

```
!
 hostname Spoke1
 !
 crypto isakmp policy 1
 encryption aes
 authentication pre-share
 group 14
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto IPsec transform-set trans2 esp-aes esp-sha-hmac
  mode transport
 !
 crypto IPsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.11 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
ip nhrp map 10.0.0.1 172.16.0.1
ip nhrp map multicast 172.16.0.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
ip nhrp nhs 10.0.0.1
ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet 0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection IPsec profile vpnprof shared
 !
 interface Tunnel1
  bandwidth 1000
  ip address 10.0.1.11 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
ip nhrp map 10.0.1.1 172.16.0.5
ip nhrp map multicast 172.16.0.5
  ip nhrp network-id 100001
  ip nhrp holdtime 300
```

```
ip nhrp nhs 10.0.1.1
ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100001
  tunnel protection IPsec profile vpnprof shared
 !
 interface Ethernet 0
  ip address dhcp hostname Spoke1
 !
 interface Ethernet1
  ip address 192.168.1.1 255.255.255.0
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
  network 10.0.1.0 0.0.0.255
  network 192.168.1.0 0.0.0.255
  no auto-summary
 !
```

Spoke 2 Configuration

Spoke 2 has the following DMVPN configuration:

```
!
 hostname Spoke2
 !
 crypto isakmp policy 1
 encryption aes
 authentication pre-share
 group 14
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto IPsec transform-set trans2 esp-aes esp-sha-hmac
  mode transport
 !
 crypto IPsec profile vpnprof
  set transform-set trans2
 !
 interface Tunnel0
  bandwidth 1000
  ip address 10.0.0.12 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
ip nhrp map 10.0.0.1 172.16.0.1
ip nhrp map multicast 172.16.0.1
  ip nhrp network-id 100000
  ip nhrp holdtime 300
ip nhrp nhs 10.0.0.1
ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet 0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection IPsec profile vpnprof shared
 !
 interface Tunnel1
  bandwidth 1000
  ip address 10.0.1.12 255.255.255.0
  ip mtu 1400
  ip nhrp authentication test
ip nhrp map 10.0.1.1 172.16.0.5
```

```
ip nhrp map multicast 172.16.0.5
  ip nhrp network-id 100001
  ip nhrp holdtime 300
ip nhrp nhs 10.0.1.1
ip tcp adjust-mss 1360
  delay 1000
  tunnel source Ethernet0
  tunnel mode gre multipoint
  tunnel key 100001
  tunnel protection IPsec profile vpnprof shared
 !
 interface Ethernet 0
  ip address dhcp hostname Spoke2
 !
 interface Ethernet1
  ip address 192.168.2.1 255.255.255.0
 !
 router eigrp 1
  network 10.0.0.0 0.0.0.255
  network 10.0.1.0 0.0.0.255
  network 192.168.2.0 0.0.0.255
  no auto-summary
 !
```

Spoke 1 Output

Spoke 1 displays the following output for its DMVPN configuration:

```
Spoke1# show ip nhrp

10.0.0.1/32 via 10.0.0.1, Tunnel0 created 00:06:52, never expire
  Type: static, Flags: used
  NBMA address: 172.16.0.1
10.0.0.12/32 via 10.0.0.12, Tunnel0 created 00:03:17, expire 00:01:52
  Type: dynamic, Flags: router
  NBMA address: 172.16.0.12
10.0.1.1/32 via 10.0.1.1, Tunnel1 created 00:13:45, never expire
  Type: static, Flags: used
  NBMA address: 172.16.0.5
10.0.1.12/32 via 10.0.1.12, Tunnel1 created 00:00:02, expire 00:04:57
  Type: dynamic, Flags: router
  NBMA address: 172.16.0.12

Spoke1# show crypto socket
```

Note There are only three crypto connections because the two NHRP sessions (10.0.0.12, Tunnel0) and (10.0.1.12, Tunnel1) are only one IPsec session, because they both have the same nonbroadcast multiaccess (NBMA) IPsec peer address.

```
Number of Crypto Socket connections 3
   Shd Peers (local/remote): 172.17.0.11
/172.17.0.12
       Local Ident  (addr/mask/port/prot): (172.16.0.11/255.255.255.255/0/47)
       Remote Ident (addr/mask/port/prot): (172.16.0.12/255.255.255.255/0/47)
       Flags: shared
       IPsec Profile: "vpnprof"
       Socket State: Open
       Client: "TUNNEL SEC" (Client State: Active)
   Shd Peers (local/remote): 172.16.0.11
```

```
/172.17.0.5
       Local Ident  (addr/mask/port/prot): (172.16.0.11/255.255.255.255/0/47)
       Remote Ident (addr/mask/port/prot): (172.16.0.5/255.255.255.255/0/47)
       Flags: shared
       IPsec Profile: "vpnprof"
       Socket State: Open
       Client: "TUNNEL SEC" (Client State: Active)
   Shd Peers (local/remote): 172.16.0.11
/172.17.0.1
       Local Ident  (addr/mask/port/prot): (172.17.0.11/255.255.255.255/0/47)
       Remote Ident (addr/mask/port/prot): (172.17.0.1/255.255.255.255/0/47)
       Flags: shared
       IPsec Profile: "vpnprof"
       Socket State: Open
       Client: "TUNNEL SEC" (Client State: Active)
Crypto Sockets in Listen state:
Client: "TUNNEL SEC" Profile: "vpnprof" Map-name: "vpnprof-head-1"

Spoke1# show crypto map

Crypto Map: "vpnprof-head-1" idb: Ethernet0/0 local address: 172.16.0.11
Crypto Map "vpnprof-head-1" 65536 IPsec-isakmp
        Profile name: vpnprof
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                trans2,
        }
Crypto Map "vpnprof-head-1" 65537 IPsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.17.0.5
        Extended IP access list
            access-list  permit gre host 172.16.0.11 host 172.16.0.5
        Current peer: 172.17.0.5
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                trans2,
        }
Crypto Map "vpnprof-head-1" 65538 IPsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.17.0.1
        Extended IP access list
            access-list  permit gre host 172.16.0.11 host 172.16.0.1
        Current peer: 172.17.0.1
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                trans2,
        }
Crypto Map "vpnprof-head-1" 65539 IPsec-isakmp
        Map is a PROFILE INSTANCE.
        Peer = 172.17.0.12
        Extended IP access list
            access-list  permit gre host 172.16.0.11 host 172.16.0.12
        Current peer: 172.17.0.12
        Security association lifetime: 4608000 kilobytes/3600 seconds
        PFS (Y/N): N
        Transform sets={
                trans2,
        }
        Interfaces using crypto map vpnprof-head-1:
                Tunnel1
                Tunnel0
```

Note All three crypto sessions are shown under each tunnel interface (three entries, twice) in the **show crypto IPsec sa** command output, because both interfaces are mapped to the same IPsec SADB, which has three entries. This duplication of output is expected in this case.

```
Spoke1# show crypto IPsec sa

interface: Tunnel0
    Crypto map tag: vpnprof-head-1, local addr 172.16.0.11
   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.0.11/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (172.16.0.1/255.255.255.255/47/0)
   current_peer 172.16.0.1 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 134, #pkts encrypt: 134, #pkts digest: 134
    #pkts decaps: 118, #pkts decrypt: 118, #pkts verify: 118
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 22, #recv errors 0
     local crypto endpt.: 172.16.0.11, remote crypto endpt.: 172.16.0.1
     path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
     current outbound spi: 0xA75421B1(2807308721)
     inbound esp sas:
      spi: 0x96185188(2518176136)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Transport, }
        conn id: 3, flow_id: SW:3, crypto map: vpnprof-head-1
        sa timing: remaining key lifetime (k/sec): (4569747/3242)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE
     inbound ah sas:
     inbound pcp sas:
     outbound esp sas:
      spi: 0xA75421B1(2807308721)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Transport, }
        conn id: 4, flow_id: SW:4, crypto map: vpnprof-head-1
        sa timing: remaining key lifetime (k/sec): (4569745/3242)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE
     outbound ah sas:
     outbound pcp sas:
   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.0.11/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (172.16.0.5/255.255.255.255/47/0)
   current_peer 172.16.0.5 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 244, #pkts encrypt: 244, #pkts digest: 244
    #pkts decaps: 253, #pkts decrypt: 253, #pkts verify: 253
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 1, #recv errors 0
     local crypto endpt.: 172.16.0.11, remote crypto endpt.: 172.16.0.5
     path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
     current outbound spi: 0x3C50B3AB(1011921835)
     inbound esp sas:
```

```
      spi: 0x3EBE84EF(1052673263)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Transport, }
        conn id: 1, flow_id: SW:1, crypto map: vpnprof-head-1
        sa timing: remaining key lifetime (k/sec): (4549326/2779)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE
     inbound ah sas:
     inbound pcp sas:
     outbound esp sas:
      spi: 0x3C50B3AB(1011921835)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Transport, }
        conn id: 2, flow_id: SW:2, crypto map: vpnprof-head-1
        sa timing: remaining key lifetime (k/sec): (4549327/2779)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE
     outbound ah sas:
     outbound pcp sas:
   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.0.11/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (172.16.0.12/255.255.255.255/47/0)
   current_peer 172.16.0.12 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
    #pkts decaps: 2, #pkts decrypt: 2, #pkts verify: 2
    #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.: 172.16.0.11, remote crypto endpt.: 172.16.0.12
     path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
     current outbound spi: 0x38C04B36(952126262)
     inbound esp sas:
      spi: 0xA2EC557(170837335)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Transport, }
        conn id: 5, flow_id: SW:5, crypto map: vpnprof-head-1
        sa timing: remaining key lifetime (k/sec): (4515510/3395)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE
     inbound ah sas:
     inbound pcp sas:
     outbound esp sas:
      spi: 0x38C04B36(952126262)
        transform: esp-aes esp-sha-hmac ,
        in use settings ={Transport, }
        conn id: 6, flow_id: SW:6, crypto map: vpnprof-head-1
        sa timing: remaining key lifetime (k/sec): (4515511/3395)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE
     outbound ah sas:
     outbound pcp sas:
interface: Tunnel1
    Crypto map tag: vpnprof-head-1, local addr 172.16.0.11
   protected vrf: (none)
   local  ident (addr/mask/prot/port): (172.16.0.11/255.255.255.255/47/0)
   remote ident (addr/mask/prot/port): (172.16.0.1/255.255.255.255/47/0)
   current_peer 172.16.0.1 port 500
     PERMIT, flags={origin_is_acl,}
```

```
 #pkts encaps: 134, #pkts encrypt: 134, #pkts digest: 134
 #pkts decaps: 118, #pkts decrypt: 118, #pkts verify: 118
 #pkts compressed: 0, #pkts decompressed: 0
 #pkts not compressed: 0, #pkts compr. failed: 0
 #pkts not decompressed: 0, #pkts decompress failed: 0
 #send errors 22, #recv errors 0
  local crypto endpt.: 172.16.0.11, remote crypto endpt.: 172.16.0.1
  path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
  current outbound spi: 0xA75421B1(2807308721)
  inbound esp sas:
   spi: 0x96185188(2518176136)
     transform: esp-aes esp-sha-hmac ,
     in use settings ={Transport, }
     conn id: 3, flow_id: SW:3, crypto map: vpnprof-head-1
     sa timing: remaining key lifetime (k/sec): (4569747/3242)
     IV size: 16 bytes
     replay detection support: Y
     Status: ACTIVE
  inbound ah sas:
  inbound pcp sas:
  outbound esp sas:
   spi: 0xA75421B1(2807308721)
     transform: esp-aes esp-sha-hmac ,
     in use settings ={Transport, }
     conn id: 4, flow_id: SW:4, crypto map: vpnprof-head-1
     sa timing: remaining key lifetime (k/sec): (4569745/3242)
     IV size: 16 bytes
     replay detection support: Y
     Status: ACTIVE
  outbound ah sas:
  outbound pcp sas:
protected vrf: (none)
local  ident (addr/mask/prot/port): (172.16.0.11/255.255.255.255/47/0)
remote ident (addr/mask/prot/port): (172.16.0.5/255.255.255.255/47/0)
current_peer 172.16.0.5 port 500
  PERMIT, flags={origin_is_acl,}
 #pkts encaps: 244, #pkts encrypt: 244, #pkts digest: 244
 #pkts decaps: 253, #pkts decrypt: 253, #pkts verify: 253
 #pkts compressed: 0, #pkts decompressed: 0
 #pkts not compressed: 0, #pkts compr. failed: 0
 #pkts not decompressed: 0, #pkts decompress failed: 0
 #send errors 1, #recv errors 0
  local crypto endpt.: 172.16.0.11, remote crypto endpt.: 172.16.0.5
  path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
  current outbound spi: 0x3C50B3AB(1011921835)
  inbound esp sas:
   spi: 0x3EBE84EF(1052673263)
     transform: esp-aes esp-sha-hmac ,
     in use settings ={Transport, }
     conn id: 1, flow_id: SW:1, crypto map: vpnprof-head-1
     sa timing: remaining key lifetime (k/sec): (4549326/2779)
     IV size: 16 bytes
     replay detection support: Y
     Status: ACTIVE
  inbound ah sas:
  inbound pcp sas:
  outbound esp sas:
   spi: 0x3C50B3AB(1011921835)
     transform: esp-aes esp-sha-hmac ,
     in use settings ={Transport, }
     conn id: 2, flow_id: SW:2, crypto map: vpnprof-head-1
     sa timing: remaining key lifetime (k/sec): (4549327/2779)
     IV size: 16 bytes
     replay detection support: Y
```

```
        Status: ACTIVE
      outbound ah sas:
      outbound pcp sas:
    protected vrf: (none)
    local  ident (addr/mask/prot/port): (172.16.0.11/255.255.255.255/47/0)
    remote ident (addr/mask/prot/port): (172.16.0.12/255.255.255.255/47/0)
    current_peer 172.16.0.12 port 500
      PERMIT, flags={origin_is_acl,}
     #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
     #pkts decaps: 2, #pkts decrypt: 2, #pkts verify: 2
     #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.: 172.16.0.11, remote crypto endpt.: 172.16.0.12
      path mtu 1500, ip mtu 1500, ip mtu idb Ethernet0/0
      current outbound spi: 0x38C04B36(952126262)
      inbound esp sas:
       spi: 0xA2EC557(170837335)
         transform: esp-aes esp-sha-hmac ,
         in use settings ={Transport, }
         conn id: 5, flow_id: SW:5, crypto map: vpnprof-head-1
         sa timing: remaining key lifetime (k/sec): (4515510/3395)
         IV size: 16 bytes
         replay detection support: Y
         Status: ACTIVE
      inbound ah sas:
      inbound pcp sas:
      outbound esp sas:
       spi: 0x38C04B36(952126262)
         transform: esp-aes esp-sha-hmac ,
         in use settings ={Transport, }
         conn id: 6, flow_id: SW:6, crypto map: vpnprof-head-1
         sa timing: remaining key lifetime (k/sec): (4515511/3395)
         IV size: 16 bytes
         replay detection support: Y
         Status: ACTIVE
      outbound ah sas:
      outbound pcp sas:
Spoke1#
```

Additional References for Sharing IPsec with Tunnel Protection

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Security commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*
Configuring DMVPN	*Dynamic Multipoint VPN (DMVPN)*

Related Topic	Document Title
Implementing DMVPN with IPsec VPN solution	Dynamic Multipoint IPsec VPNs (Using Multipoint GRE/NHRP to Scale IPsec VPNs)
Configuring basic IPsec VPNs	*Configuring Security for VPNs with IPsec*
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2547	*BGP/MPLS VPNs*
RFC 2784	*Generic Routing Encapsulation (GRE)*

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Sharing IPsec with Tunnel Protection

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1: Feature Information for Sharing IPsec with Tunnel Protection

Feature Name	Releases	Feature Information
Sharing IPsec with Tunnel Protection	12.4(15)T	The Sharing IPsec with Tunnel Protection feature allows sharing an IPsec security association database (SADB) between two or more generic routing encapsulation (GRE) tunnel interfaces when tunnel protection is used. Shared tunnel interfaces have a single underlying cryptographic SADB, cryptographic map, and IPsec profile in the Dynamic Multipoint Virtual Private Network (DMVPN) configuration. The Sharing IPsec with Tunnel Protection feature is required in some DMVPN configurations. If IPsec SA sessions are not shared within the same IPsec SADB, an IPsec SA may be associated with the wrong IPsec SADB and therefore with the wrong tunnel interface, thereby causing duplicate IPsec security associations (SAs) and tunnel interfaces to flap, which in turn results in network connectivity problems. The following command was introduced or modified: **tunnel protection IPsec profile**.

Glossary

GRE—generic routing encapsulation. Tunnels that provide a specific pathway across the shared WAN and encapsulate traffic with new packet headers to ensure delivery to specific destinations. The network is private because traffic can enter a tunnel only at an endpoint. Tunnels do not provide true confidentiality (encryption does) but can carry encrypted traffic.

GRE tunneling can also be used to encapsulate non-IP traffic into IP and send it over the Internet or IP network. The Internet Package Exchange (IPX) and AppleTalk protocols are examples of non-IP traffic.

IKE—Internet Key Exchange. A hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the ISAKMP framework. Although IKE can be used with other protocols, its initial implementation is with IPsec. IKE provides authentication of the IPsec peers, negotiates IPsec keys, and negotiates IPsec security associations.

IPsec—IP security. A framework of open standards developed by the Internet Engineering Task Force (IETF). IPsec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec peers, such as Cisco routers.

ISAKMP—Internet Security Association Key Management Protocol. A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.

NHRP—Next Hop Resolution Protocol. A protocol that routers, access servers, and hosts can use to discover the addresses of other routers and hosts connected to an NBMA network.

The Cisco implementation of NHRP supports the IETF draft version 11 of NBMA NHRP.

The Cisco implementation of NHRP supports IP Version 4, Internet Packet Exchange (IPX) network layers, and, at the link layer, ATM, Ethernet, SMDS, and multipoint tunnel networks. Although NHRP is available on Ethernet, NHRP need not be implemented over Ethernet media because Ethernet is capable of broadcasting. Ethernet support is unnecessary (and not provided) for IPX.

SA—security association. Describes how two or more entities use security services to communicate securely. For example, an IPsec SA defines the encryption algorithm (if used), the authentication algorithm, and the shared session key to be used during the IPsec connection.

Both IPsec and IKE require and use SAs to identify the parameters of their connections. IKE can negotiate and establish its own SA. The IPsec SA is established either by IKE or by manual user configuration.

transform—List of operations performed on a data flow to provide data authentication, data confidentiality, and data compression. An example of a transform is the ESP with the 256-bit AES encryption algorithm and the AH protocol with the HMAC-SHA authentication algorithm.

tunnel—In the context of this module, a secure communication path between two peers, such as two routers. It does not refer to using IPsec in tunnel mode.

VPN—Virtual Private Network. A framework that consists of multiple peers transmitting private data securely to one another over an otherwise public infrastructure. In this framework, inbound and outbound network traffic is protected using protocols that tunnel and encrypt all data. This framework permits networks to extend beyond their local topology, while remote users are provided with the appearance and functionality of a direct network connection.