Sharing IPsec with Tunnel Protection

The Sharing IPsec with Tunnel Protection feature allows an IP Security (IPsec) Security Association Database (SADB) to be shared between two or more generic routing encapsulation (GRE) tunnel interfaces when tunnel protection is used. These tunnel interfaces share a single underlying cryptographic SADB, cryptographic map, and IPsec profile in the Dynamic Multipoint Virtual Private Network (DMVPN) configuration.

If IPsec security association (SA) sessions are not shared in the same IPsec SADB, then an IPsec SA may get associated with an undesired IPsec SADB, and may also get associated with a wrong tunnel interface, causing duplication of IPsec SAs and flapping of tunnel interfaces. If the tunnel interfaces flap (change rapidly and repeatedly between online and offline states), then network connectivity problems occur.

Note Security threats, as well as the cryptographic technologies to help protect against them, 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.

Prerequisites for Sharing IPsec with Tunnel Protection

- Before a multipoint GRE (mGRE) and IPsec tunnel can be established, you must define an Internet Key Exchange (IKE) policy by using the **crypto isakmp policy** command.

Restrictions for Sharing IPsec with Tunnel Protection

- The **tunnel source** command on all the tunnel interfaces that use the same tunnel source must be configured using interface type and number, not the tunnel's IP address.
- All tunnels with the same tunnel source interface must use the same IPsec profile and must have the **tunnel protection shared** command configured. The only exception is a scenario when there are only peer-to-peer (P2P) GRE tunnel interfaces configured with the same tunnel source in the system, all with unique tunnel destination IP addresses.
- Different IPsec profile names must be used for shared and unshared tunnels.

  For example, if “tunnel 1” is configured with the **tunnel source loopback0** command, and “tunnel 2” and “tunnel 3” are shared using the **tunnel source loopback1** command, use ipsec-profile-1 for tunnel 1 and ipsec-profile-2 for tunnels 2 and 3.
- A different IPsec profile must be used for each set of shared tunnels.

  For example, if tunnels 1 through 5 use **loopback0** as their tunnel source and tunnels 6 through 10 use loopback1, then define the profile ipsec-profile-1 for tunnels 1 through 5 and ipsec-profile-2 for tunnels 6 through 10.
- It may be desirable to not 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. For 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 use the ISAKMP profile to connect to an IPsec profile. This ISAKMP profile limits the IPsec profile to accept only those connections that matched the corresponding ISAKMP profile. Separate ISAKMP and IPsec profiles can be obtained for each DMVPN cloud (tunnel interface) without sharing the same IPsec SADB.
- Sharing IPsec is not desired and not supported for a virtual tunnel interface (VTI). A VTI provides a routable interface type for terminating IPsec tunnels and a way to define protection between sites to form an overlay network.
- Sharing IPsec is not supported on Virtual-Template type tunnel interfaces. It cannot be used either in the default **tunnel mode gre ip** mode with IPsec protection, (for example, FlexVPN) or with the **tunnel mode ipsec ipv4** (for example, Dynamic Virtual Tunnel interface - DVTI). Each virtual-template interface must have a separate and unshared IPsec profile. Otherwise, the router might crash after the virtual-access is deleted.

Information About Sharing IPsec with Tunnel Protection

Single IPsec SAs and GRE Tunnel Sessions

In a dual-hub, dual-DMVPN topology, it is possible to have two or more GRE tunnel sessions (same tunnel source and destination, but different tunnel keys) between the two endpoints of the same type. In this case, you should use a single IPsec SA to secure both GRE tunnel sessions. It is not possible to determine the tunnel interface under which 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 configuration allows a single IPsec SA to be used for all GRE tunnels (same tunnel source and destination, but different tunnel keys) between two endpoints of the same type. The **tunnel protection ipsec profile shared** command 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 not shared).

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

When the IPsec path maximum transmission unit (MTU) changes, the value of SA MTU in the Quantum Flow Processor (QFP) and the hardware cryptographic engine gets updated and becomes consistent with the IPsec MTU. While the MTU changes, the system may drop some packets and transient %ATTN-3-SYNC_TIMEOUT errors may be displayed on the console.

Note The tunnel source, tunnel destination, and tunnel key (triplet) must be unique for all tunnel interfaces on a router. For a multipoint GRE (mGRE) interface where the tunnel destination is not configured, the pair (tunnel source and tunnel key) must be unique. Incoming GRE packets are also matched to P2P GRE tunnels first; if there is no match, then they are matched to mGRE tunnels.

How to Configure Sharing IPsec with Tunnel Protection

Sharing an IPsec SADB Between Multiple Tunnel Interfaces in a DMVPN

Perform this task to configure a Cisco IOS router to share an IPsec SADB between multiple tunnel interfaces in a DMVPN.

If your configuration requires more spoke routers in a dual-hub, dual DMVPN topology, repeat the steps listed in this task to configure additional spokes.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **interface tunnel** *number*
4. **tunnel source** {*ip-address* | *interface-type interface-number*}
5. **tunnel protection ipsec profile** *name* [**shared]**
6. **exit**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(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 you can create.
Step 4	**tunnel source** {*ip-address* | *interface-type interface-number*} **Example:** `Router(config-if)# tunnel source GigabitEthernet 0`	Sets the source IP address or source interface type number for a tunnel interface. • When you are using the **tunnel protection ipsec profile** command, you must specify an interface, not an IP address for the tunnel source.
Step 5	**tunnel protection ipsec profile** *name* [**shared]** **Example:** `Router(config-if)# tunnel protection ipsec profile vpnprof shared`	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. • The **shared** keyword allows IPsec sessions to be shared between multiple tunnel interfaces configured with the same tunnel source IP.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 7	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and enters privileged EXEC mode.

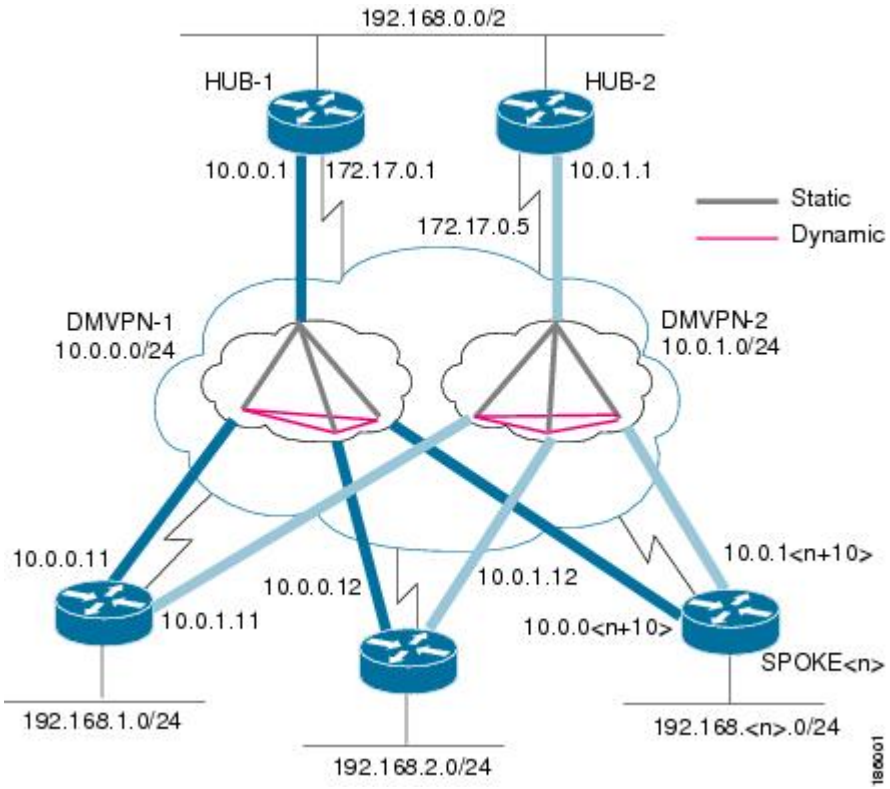
Configuration Examples for Sharing IPsec with Tunnel Protection

Example: Dual-Hub Router, Dual-DMVPN Topology

The dual-hub router, dual-DMVPN topology, shown in the following figure, has the following attributes:

- Each hub router is configured with a single mGRE tunnel interface.
- Each hub router is connected to one DMVPN subnet (cloud), and the spokes are connected to both DMVPN-1 and DMVPN-2.
- Each spoke router 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, Dual-DMVPN Topology



Example: Configuring an IPsec SADB Between Multiple Tunnel Interfaces in a DMVPN

Example: HUB-1 Configuration

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 Address Resolution Protocol (NHRP) network ID: 100000
- Tunnel key: 100000
- Dynamic routing protocol: Enhanced Interior Gateway Routing Protocol (EIGRP)

```
!
hostname Hub1
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto IPsec transform-set trans2 esp-des esp-md5-hmac
```

```
 mode transport
!
crypto IPsec profile vpnprof
 set transform-set trans2
!
interface Tunnel 5
 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 GigabitEthernet 0/0/0
  tunnel mode gre multipoint
  tunnel key 100000
  tunnel protection IPsec profile vpnprof
!
interface GigabitEthernet 0/0/0
 ip address 172.17.0.1 255.255.255.252
!
interface GigabitEthernet 0/0/1
 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
!
```

Example: 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
 authentication pre-share
crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
!
crypto ipsec transform-set trans2 esp-des esp-md5-hmac
 mode transport
!
crypto ipsec profile vpnprof
 set transform-set trans2
!
interface Tunnel 5
 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 GigabitEthernet 0/0/0
  tunnel mode gre multipoint
  tunnel key 100001
  tunnel protection ipsec profile vpnprof
 !
 interface GigabitEthernet 0/0/0
  ip address 172.17.0.5 255.255.255.252
 !
 interface GigabitEthernet 0/0/1
  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
 !
```

Example: SPOKE 1 Configuration

SPOKE 1 has the following DMVPN configuration:

```
!
 hostname Spoke1
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
interface Tunnel 5
 bandwidth 1000
.
.
.
 ip nhrp authentication test
 ip nhrp map 10.0.0.1 172.17.0.1
 ip nhrp map multicast 172.17.0.1
 ip nhrp network-id 100000
 ip nhrp holdtime 300|
 ip nhrp nhs 10.0.0.1
 ip tcp adjust-mss 1360
 delay 1000
.
.
.
 tunnel protection ipsec profile vpnprof shared
!
interface Tunnel 5
 bandwidth 1000
```

```
.
.
.
 ip nhrp authentication test
 ip nhrp map 10.0.1.1 172.17.0.5
 ip nhrp map multicast 172.17.0.5
 ip nhrp network-id 100001
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.1.1
 ip tcp adjust-mss 1360
 delay 1000
.
.
.
 tunnel protection ipsec profile vpnprof shared
!
 interface GigabitEthernet 0/0/0
  ip address dhcp hostname Spoke1
 !
 interface GigabitEthernet 0/0/1
  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
 !
```

Example: SPOKE 2 Configuration

SPOKE 2 has the following DMVPN configuration:

```
!
 hostname Spoke2
 !
 crypto isakmp policy 1
  authentication pre-share
 crypto isakmp key cisco47 address 0.0.0.0 0.0.0.0
 !
 crypto ipsec transform-set trans2 esp-des esp-md5-hmac
  mode transport
 !
 crypto ipsec profile vpnprof
  set transform-set trans2
 !
interface Tunnel 5
 bandwidth 1000
.
.
.
 ip nhrp authentication test
 ip nhrp map 10.0.0.1 172.17.0.1
 ip nhrp map multicast 172.17.0.1
 ip nhrp network-id 100000
 ip nhrp holdtime 300|
 ip nhrp nhs 10.0.0.1
 ip tcp adjust-mss 1360
 delay 1000
.
.
.
 tunnel protection ipsec profile vpnprof shared
```

```
!
interface Tunnel 5
 bandwidth 1000
.
.
.
 ip nhrp authentication test
 ip nhrp map 10.0.1.1 172.17.0.5
 ip nhrp map multicast 172.17.0.5
 ip nhrp network-id 100001
 ip nhrp holdtime 300
 ip nhrp nhs 10.0.1.1
 ip tcp adjust-mss 1360
 delay 1000
.
.
.
 tunnel protection ipsec profile vpnprof shared
 !
 interface GigabitEthernet 0/0/0
  ip address dhcp hostname Spoke2
 !
 interface GigabitEthernet 0/0/1
  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
 !
```

Example: Results on SPOKE 1

SPOKE 1 has the following results for its DMVPN configuration:

```
Spoke1# show ip nhrp

10.0.0.1/32 via 10.0.0.1, Tunnel 0 created 00:06:52, never expire
  Type: static, Flags: used
  NBMA address: 172.17.0.1
10.0.0.12/32 via 10.0.0.12, Tunnel 0 created 00:03:17, expire 00:01:52
  Type: dynamic, Flags: router
  NBMA address: 172.17.0.12
10.0.1.1/32 via 10.0.1.1, Tunnel 1 created 00:13:45, never expire
  Type: static, Flags: used
  NBMA address: 172.17.0.5
10.0.1.12/32 via 10.0.1.12, Tunnel 1 created 00:00:02, expire 00:04:57
  Type: dynamic, Flags: router
  NBMA address: 172.17.0.12
Spoke1# show crypto socket
```

Note There are only three crypto connections (172.17.0.12, 172.17.0.5 and 172.17.0.1). The two NHRP sessions (10.0.0.12, Tunnel 0) and (10.0.1.12, Tunnel 1) represent the same 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): (172.17.0.11/255.255.255.255/0/47)
       Remote Ident (addr/mask/port/prot): (172.17.0.12/255.255.255.255/0/47)
       Flags: shared
       ipsec Profile: "vpnprof"
       Socket State: Open
       Client: "TUNNEL SEC" (Client State: Active)
   Shd Peers (local/remote): 172.17.0.11
/172.17.0.5
       Local Ident  (addr/mask/port/prot): (172.17.0.11/255.255.255.255/0/47)
       Remote Ident (addr/mask/port/prot): (172.17.0.5/255.255.255.255/0/47)
       Flags: shared
       ipsec Profile: "vpnprof"
       Socket State: Open
       Client: "TUNNEL SEC" (Client State: Active)
   Shd Peers (local/remote): 172.17.0.11
/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: FastEthernet0/0/0 local address: 172.17.0.11
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 172.17.0.11 host 172.17.0.5
        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 172.17.0.11 host 172.17.0.1
        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 172.17.0.11 host 172.17.0.12
        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 The three crypto sessions are shown under both tunnel interface (three entries, twice) in the **show crypto ipsec sa** 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: Tunnel 0
   Crypto map tag: vpnprof-head-1, local addr 172.17.0.11
   protected vrf: (none)
           local  ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
           remote ident (addr/mask/prot/port): (172.17.0.1/255.255.255.255/47/0)
           current_peer 172.17.0.1 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.: 172.17.0.11, remote crypto endpt.: 172.17.0.1
     path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0/0
     current outbound spi: 0xA75421B1(2807308721)
     inbound esp sas:
      spi: 0x96185188(2518176136)
        transform: esp-des esp-md5-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: 8 bytes
        replay detection support: Y
        Status: ACTIVE
     inbound ah sas:
     inbound pcp sas:
     outbound esp sas:
      spi: 0xA75421B1(2807308721)
        transform: esp-des esp-md5-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: 8 bytes
        replay detection support: Y
        Status: ACTIVE
     outbound ah sas:
     outbound pcp sas:
   protected vrf: (none)
           local  ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
           remote ident (addr/mask/prot/port): (172.17.0.5/255.255.255.255/47/0)
           current_peer 172.17.0.5 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.: 172.17.0.11, remote crypto endpt.: 172.17.0.5
  path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0/0
  current outbound spi: 0x3C50B3AB(1011921835)
  inbound esp sas:
   spi: 0x3EBE84EF(1052673263)
     transform: esp-des esp-md5-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: 8 bytes
     replay detection support: Y
     Status: ACTIVE
  inbound ah sas:
  inbound pcp sas:
  outbound esp sas:
   spi: 0x3C50B3AB(1011921835)
     transform: esp-des esp-md5-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: 8 bytes
     replay detection support: Y
     Status: ACTIVE
  outbound ah sas:
  outbound pcp sas:
protected vrf: (none)
        local  ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
        remote ident (addr/mask/prot/port): (172.17.0.12/255.255.255.255/47/0)
        current_peer 172.17.0.12 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.: 172.17.0.11, remote crypto endpt.: 172.17.0.12
  path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0/0
  current outbound spi: 0x38C04B36(952126262)
  inbound esp sas:
   spi: 0xA2EC557(170837335)
     transform: esp-des esp-md5-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: 8 bytes
     replay detection support: Y
     Status: ACTIVE
  inbound ah sas:
  inbound pcp sas:
  outbound esp sas:
   spi: 0x38C04B36(952126262)
     transform: esp-des esp-md5-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: 8 bytes
     replay detection support: Y
     Status: ACTIVE
  outbound ah sas:
```

```
  outbound pcp sas:
     interface: Tunnel 1
     Crypto map tag: vpnprof-head-1, local addr 172.17.0.11
protected vrf: (none)
        local  ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
        remote ident (addr/mask/prot/port): (172.17.0.1/255.255.255.255/47/0)
        current_peer 172.17.0.1 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.: 172.17.0.11, remote crypto endpt.: 172.17.0.1
  path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0/0
  current outbound spi: 0xA75421B1(2807308721)
  inbound esp sas:
   spi: 0x96185188(2518176136)
     transform: esp-des esp-md5-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: 8 bytes
     replay detection support: Y
     Status: ACTIVE
  inbound ah sas:
  inbound pcp sas:
  outbound esp sas:
   spi: 0xA75421B1(2807308721)
     transform: esp-des esp-md5-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: 8 bytes
     replay detection support: Y
     Status: ACTIVE
  outbound ah sas:
  outbound pcp sas:
protected vrf: (none)
        local  ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
        remote ident (addr/mask/prot/port): (172.17.0.5/255.255.255.255/47/0)
        current_peer 172.17.0.5 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.: 172.17.0.11, remote crypto endpt.: 172.17.0.5
  path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0/0
  current outbound spi: 0x3C50B3AB(1011921835)
  inbound esp sas:
   spi: 0x3EBE84EF(1052673263)
     transform: esp-des esp-md5-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: 8 bytes
     replay detection support: Y
     Status: ACTIVE
  inbound ah sas:
  inbound pcp sas:
```

```
  outbound esp sas:
   spi: 0x3C50B3AB(1011921835)
     transform: esp-des esp-md5-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: 8 bytes
     replay detection support: Y
     Status: ACTIVE
  outbound ah sas:
  outbound pcp sas:
protected vrf: (none)
       local  ident (addr/mask/prot/port): (172.17.0.11/255.255.255.255/47/0)
       remote ident (addr/mask/prot/port): (172.17.0.12/255.255.255.255/47/0)
       current_peer 172.17.0.12 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.: 172.17.0.11, remote crypto endpt.: 172.17.0.12
  path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0/0
  current outbound spi: 0x38C04B36(952126262)
  inbound esp sas:
   spi: 0xA2EC557(170837335)
     transform: esp-des esp-md5-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: 8 bytes
     replay detection support: Y
     Status: ACTIVE
  inbound ah sas:
  inbound pcp sas:
  outbound esp sas:
   spi: 0x38C04B36(952126262)
     transform: esp-des esp-md5-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: 8 bytes
     replay detection support: Y
     Status: ACTIVE
  outbound ah sas:
  outbound pcp sas:
```

Additional References

Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	*IPv6 Configuration Guide*
Dynamic Multipoint VPN	*Dynamic Multipoint VPN Configuration Guide*

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
IPv6 commands	*IPv6 Command Reference*
Cisco IOS IPv6 features	IPv6 Feature Mapping
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	*IPv6 RFcs*

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	Cisco IOS XE Release 2.5	The Sharing IPsec with Tunnel Protection feature allows an IPsec session to be shared between two or more GRE tunnel interfaces. In Cisco IOS XE Release 2.5, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. The following command was modified by this feature: **tunnel protection ipsec profile shared.**

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 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. 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, 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. For example, one transform is the Encapsulating Security Payload (ESP) protocol with

the Hash-based Message Authentication Code (HMAC)-Message Digest Algorithm (MD5) authentication algorithm; another transform is the Authentication Header (AH) protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-Secure Hash Algorithm (SHA) authentication algorithm.

tunnel—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.