Configuring GRE over IPsec

Restrictions for GRE over IPsec

- GRE over IPsec doesn't support Multipoint GRE (mGRE).
- GRE over IPSec doesn't support multiple sessions from the same tunnel source to the same tunnel destination.
- GRE over IPsec doesn't support concurrent Static Virtual Tunnel Interface (SVTI) and GRE over IPsec tunnel with the same tunnel source and tunnel destination.
- Keepalive is not supported on VRF aware GRE over IPsec tunnels.

Information about GRE Over IPsec

You can configure Generic Routing Encapsulation (GRE) over an Internet Protocol Security (IPsec) tunnel on Cisco IOS XE devices. GRE can encapsulate several types of traffic such as unicast, multicast, broadcast, and MPLS. However, GRE doesn't provide any type of protection for the transmitted payload.

Internet Protocol Security (IPsec) provides confidentiality, integrity, and authentication to the payloads transmitted through IPsec tunnels. However, IPsec can function only with IP packets.

The GRE over IPsec feature allows for the flexibility of using GRE along with the security of IPsec. GRE encapsulates the packets. IPsec encrypts the packets and transports them through an IPsec tunnel.

Overview of VRF aware GRE over IPsec

A VRF instance is a per-VPN routing information repository that defines the VPN membership of a customer site attached to the Provider Edge (PE) router. A VRF comprises an IP routing table, a derived Cisco Express Forwarding (CEF) table, a set of interfaces that use the forwarding table, and a set of rules and routing protocol

parameters that control the information that is included in the routing table. A separate set of routing and CEF tables is maintained for each VPN customer.

Starting with the 17.13.1 release, VRF support has been introduced for GRE over IPsec tunnels.

Each GRE over IPsec tunnel is associated with two VRF domains. The outer encapsulated packet belongs to one VRF domain, which is called Front Door VRF (FVRF), while the inner, protected IP packet belongs to another domain called the Inside VRF (IVRF). The local endpoint of the IPsec tunnel belongs to the FVRF while the source and destination addresses of the inside packet belong to the IVRF.

In a VRF aware GRE over IPsec instance, when a packet starts from the service provider network it gets encapsulated based on the security policy. The encapsulated packet is forwarded using the FVRF routing table. When the packet arrives at the PE endpoint the security associations are validated. Then the packet is decapsulated and associated with an IVRF. The packet is forwarded using the IVRF routing table.

One or more IPsec tunnels can terminate on a single interface. The FVRF of all these tunnels is the same and is set to the VRF that is configured on that interface. The IVRF of these tunnels can be different and depends on the VRF that is defined in the Internet Security Association and Key Management Protocol (ISAKMP) profile that is attached to a crypto map entry.

How to Configure GRE over IPsec

The following sections explain the procedures that you can perform to configure a GRE over IPsec tunnel interface.

Configuring the IKEv2 Keyring

Perform this task to configure the IKEv2 keyring if the local or remote authentication method is a preshared key.

Configure the IKEv2 keyring keys in the peer configuration submode that defines a peer subblock. An IKEv2 keyring can have multiple peer subblocks. A peer subblock contains a single symmetric or asymmetric key pair for a peer or peer group. Any combination of the hostname, identity, and IP address identifies the peer or the peer group.

IKEv2 keyrings are independent of IKEv1 keyrings. The key differences are as follows:

- IKEv2 keyrings support symmetric and asymmetric preshared keys.
- IKEv2 keyrings don't support Rivest, Shamir, and Adleman (RSA) public keys.
- IKEv2 keyrings are specified in the IKEv2 profile and aren’t looked up, unlike IKEv1 keys. IKEv1 keys are looked up on receipt of MM1 to negotiate the preshared key authentication method. IKEv2 doesn't negotiate the authentication method.
- IKEv2 keyrings aren't associated with VPN routing and forwarding (VRF) during configuration. The VRF of an IKEv2 keyring is the VRF of the IKEv2 profile that refers to the keyring.
- You can specify a single keyring in an IKEv2 profile, unlike an IKEv1 profile, which can specify multiple keyrings.
- If peers matching different profiles share the same keys, you can specify a single keyring in more than one IKEv2 profile, .
- An IKEv2 keyring is structured as one or more peer subblocks.

On an IKEv2 initiator, the IKEv2 keyring key lookup is performed using the hostname or the address of the peer, in that order. On an IKEv2 responder, the key lookup is performed using the IKEv2 identity or the address of the peer, in that order.

Note You can't configure the same identity in more than one peer.

Procedure

	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	**crypto ikev2 keyring** *keyring-name* **Example:** `Device(config)# crypto ikev2 keyring kyr1`	Defines an IKEv2 keyring. Enters IKEv2 keyring configuration mode.
Step 4	**peer** *name* **Example:** `Device(config-ikev2-keyring)# peer peer1`	Defines the peer or peer group. Enters IKEv2 keyring peer configuration mode.
Step 5	**description** *line-of-description* **Example:** `Device(config-ikev2-keyring-peer)# description this is the first peer`	(Optional) Describes the peer or peer group.
Step 6	**hostname** *name* **Example:** `Device(config-ikev2-keyring-peer)# hostname host1`	Specifies the peer using a hostname.
Step 7	**address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} **Example:** `Device(config-ikev2-keyring-peer)# address 10.0.0.1 255.255.255.0`	Specifies an IPv4 or IPv6 address or range for the peer. **Note** This IP address is the IKE endpoint address and is independent of the identity address.
Step 8	**identity** {**address** {*ipv4-address* | *ipv6-address*} | **fqdn domain** *domain-name* | **email domain** *domain-name* | **key-id** *key-id*}	Identifies the IKEv2 peer through the following identities: • E-mail

	Command or Action	Purpose

	Example:
	`Device(config-ikev2-keyring-peer)# identity address 10.0.0.5`
		• Fully qualified domain name (FQDN)
		Note When you use FQDN to identify the peer in the keyring configuration, use the IP address of the peer along with the FQDN
		`crypto ikev2 keyring key1`
		`peer headend-1`
		`address 10.1.1.1 >>>>>>>>>`
		`identity fqdn`
		`NFVIS-headend-1.cisco.com`
		`pre-shared-key Cisco123`
		• IPv4 or IPv6 address
		• Key ID
		Note The identity is available for key lookup on the IKEv2 responder only.

Step 9	**pre-shared-key** {**local** | **remote**} [**0** | **6**] *line* **hex** *hexadecimal-string* **Example:** `Device(config-ikev2-keyring-peer)# pre-shared-key local key1`	Specifies the preshared key for the peer.
Step 10	**end** **Example:** `Device(config-ikev2-keyring-peer)# end`	Exits IKEv2 keyring peer configuration mode. Returns to privileged EXEC mode.

Configuring an IKEv2 Profile (Basic)

Perform this task to configure the mandatory commands for an IKEv2 profile.

An IKEv2 profile is a repository of nonnegotiable parameters of the IKE security association (SA) (such as local or remote identities and authentication methods) and services available to authenticated peers that match the profile. An IKEv2 profile must be configured and associated with a crypto map. Use the **set ikev2-profile** *profile-name* command to associate a profile with a crypto map. To disassociate the profile, use the **no** form of the command.

The following rules apply to match statements:

- An IKEv2 profile must contain a match identity or a match certificate statement; otherwise, the profile is considered incomplete and is not used. An IKEv2 profile can have more than one match identity or match certificate statements.
- An IKEv2 profile must have a single match Front Door VPN routing and forwarding (FVRF) statement.
- When a profile is selected, multiple match statements of the same type are logically ORed, and multiple match statements of different types are logically ANDed.

- The match identity and match certificate statements are considered to be the same type of statements and are ORed.
- Configuration of overlapping profiles is considered a misconfiguration. In the case of multiple profile matches, no profile is selected.

Use the **show crypto ikev2 profile** *profile-name* command to display the IKEv2 profile.

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile profile1`	Defines an IKEv2 profile and enters the IKEv2 profile configuration mode.
Step 4	**description** *line-of-description* **Example:** `Device(config-ikev2-profile)# description This is an IKEv2 profile`	(Optional) Describes the profile.
Step 5	**aaa accounting** {**psk** | **cert** | **eap**} *list-name* **Example:** `Device(config-ikev2-profile)# aaa accounting eap list1`	(Optional) Enables authentication, authorization, and accounting (AAA) method lists for IPsec sessions. **Note** If the **psk**, **cert**, or **eap** keyword is not specified, the AAA accounting method list is used irrespective of the peer authentication method.
Step 6	**authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}} **Example:** `Device(config-ikev2-profile)# authentication local ecdsa-sig`	Specifies the local or remote authentication method. • **rsa-sig**—Specifies RSA-sig as the authentication method. • **pre-share**—Specifies the preshared key as the authentication method. • **ecdsa-sig**—Specifies ECDSA-sig as the authentication method.

	Command or Action	Purpose
		• **eap**—Specifies EAP as the remote authentication method. • **query-identity**—Queries the EAP identity from the peer. • **timeout** *seconds*—Specifies the duration, in seconds, to wait for the next IKE_AUTH request after sending the first IKE_AUTH response. **Note** You can specify only one local authentication method but multiple remote authentication methods.
Step 7	**dpd** *interval retry-interval* {**on-demand** | **periodic**} **Example:** `Device(config-ikev2-profile)# dpd 30 6 on-demand`	This step is optional. Configures Dead Peer Detection (DPD) globally for peers matching the profile. By default, the Dead Peer Detection (DPD) is disabled.
Step 8	**dynamic** **Example:** `Device(config-ikev2-profile)# dynamic`	Configures a dynamic IKEv2 profile. **Note** When you configure a dynamic profile, you cannot configure local or remote authentication and identity using the command line interface.
Step 9	**identity local** {**address** {*ipv4-address* | *ipv6-address*} | **dn** | **email** *email-string* | **fqdn** *fqdn-string* | **key-id** *opaque-string*} **Example:** `Device(config-ikev2-profile)# identity local email abc@example.com`	This is an optional step. Specifies the local IKEv2 identity type. **Note** If the local authentication method is a preshared key, the default local identity is the IP address. If the local authentication method is a Rivest, Shamir, and Adleman (RSA) signature, the default local identity is a Distinguished Name.
Step 10	**initial-contact force** **Example:** `Device(config-ikev2-profile)# initial-contact force`	Enforces initial contact processing if the initial contact notification is not received in the IKE_AUTH exchange.
Step 11	**ivrf** *name* **Example:** `Device(config-ikev2-profile)# ivrf vrf1`	This is an optional step. Specifies a user-defined VPN routing and forwarding (VRF) or global VRF if the IKEv2 profile is attached to a crypto map. • If you use the IKEv2 profile for tunnel protection, you must configure the Inside

	Command or Action	Purpose
		VRF (IVRF) for the tunnel interface on the tunnel interface. **Note** IVRF specifies the VRF for cleartext packets. The default value for IVRF is FVRF.
Step 12	**keyring** {**local** *keyring-name* | **aaa** *list-name* [**name-mangler** *mangler-name* | **password** *password*] } **Example:** `Device(config-ikev2-profile)# keyring aaa keyring1 name-mangler mangler1`	Specifies the local or AAA-based key ring that must be used with the local and remote preshared key authentication method. **Note** You can specify only one key ring. Local AAA is not supported for AAA-based preshared keys. **Note** When using AAA, the default password for a Radius access request is "cisco". You can use the **password** keyword within the **keyring** command to change the password.
Step 13	**lifetime** *seconds* **Example:** `Device(config-ikev2-profile)# lifetime 1000`	Specifies the lifetime, in seconds, for the IKEv2 SA.
Step 14	**match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map* | **fvrf** {*fvrf-name* | **any**} | **identity remote address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | {**email** [*domain string*] | **fqdn** [*domain string*]} *string* | **key-id** *opaque-string*} **Example:** `Device(config-ikev2-profile)# match address local interface Ethernet 2/0`	Uses match statements to select an IKEv2 profile for a peer.
Step 15	**pki trustpoint** *trustpoint-label* [**sign** | **verify**] **Example:** `Device(config-ikev2-profile)# pki trustpoint tsp1 sign`	Specifies Public Key Infrastructure (PKI) trustpoints for use with the RSA signature authentication method. **Note** If the **sign** or **verify** keyword is not specified, the trustpoint is used for signing and verification.

	Command or Action	Purpose
		Note In contrast to IKEv1, a trustpoint must be configured in an IKEv2 profile for certificate-based authentication to succeed. There is no fallback for globally configured trustpoints if this command is not present in the configuration. The trustpoint configuration applies to the IKEv2 initiator and responder.
Step 16	**virtual-template** *number* **mode auto** **Example:** `Device(config-ikev2-profile)# virtual-template 1 mode auto`	This is an optional step. Specifies the virtual template for cloning a virtual access interface (VAI). • **mode auto** - Enables the tunnel mode auto selection feature.
Step 17	**shutdown** **Example:** `Device(config-ikev2-profile)# shutdown`	(Optional) Shuts down the IKEv2 profile.
Step 18	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits the IKEv2 profile configuration mode and returns to the privileged EXEC mode.

Attaching an IKEv2 profile to an IPsec profile

To attach an IKEv2 profile to an IPsec profile, perform the following procedure.

Procedure

	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	**crypto ipsec transform-set** *transform-set-name* **Example:** `Device(config)# crypto ipsec transform-set tfs`	Defines a transform set. Enters crypto transform configuration mode.

	Command or Action	Purpose
Step 4	**mode tunnel** **Example:** `Device(cfg-crypto-tran)# mode tunnel`	(Optional) Changes the mode associated with the transform set.
Step 5	**crypto IPsec profile** *profile-name* **Example:** `Device(cfg-crypto-tran)# crypto IPsec profile PROF`	Defines the IPsec parameters used for IPsec encryption between two IPsec devices. Enters IPsec profile configuration mode.
Step 6	**set transform-set** *transform-set-name* **Example:** `Device(ipsec-profile)# set transform-set tfs esp-gcm`	Specifies the transform sets used with the crypto map entry.
Step 7	**set ikev2-profile** *profile-name* **Example:** `Device(ipsec-profile)# set ikev2-profile ikev2_prof`	Attaches an IKEv2 profile to an IPSec profile.
Step 8	**exit** **Example:** `Device(ipsec-profile)# exit`	Exits IPsec profile configuration mode. Enters global configuration mode.

Configuring a GRE over IPsec Tunnel Interface

To create a GRE over IPsec tunnel and configure a tunnel source and tunnel destination under the tunnel interface, perform the following procedure:

Procedure

	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.

	Command or Action	Purpose
Step 3	**interface** *tunnel number* **Example:** `Device(config)# interface tunnel 100`	Specifies the interface on which the tunnel will be configured. Enters interface configuration mode.
Step 4	**ip address** *address mask* **Example:** `Device(config-if)# ip address 128.1.1.1 255.255.255.0`	Specifies the IP address and mask.
Step 5	**tunnel source** *interface-type interface-number* **Example:** `Device(config-if)# tunnel source 120.1.1.1`	Specifies the tunnel source as a loopback interface.
Step 6	**tunnel destination** *ip-address* **Example:** `Device(config-if)# tunnel destination 120.1.1.2`	Identifies the IP address of the tunnel destination.
Step 7	**tunnel protection IPsec profile** *profile-name* **Example:** `Device(config-if)# tunnel protection IPsec profile ipsec-prof`	Associates a tunnel interface with an IPsec profile.
Step 8	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode. Returns to privileged EXEC mode.

Configuration Examples for GRE over IPsec

The following sections provide configuration examples for GRE over IPsec.

Example: Configuring GRE over IPsec

The following examples show how to configure a GRE over IPsec tunnel.

The following example shows how to configure an Internet Key Exchange Version 2 (IKEv2) key ring with symmetric preshared keys based on an IP address:

```
Device(config)# crypto ikev2 keyring ikev2_key
```

```
Device(config-ikev2-keyring)# peer mypeer
Device(config-ikev2-keyring-peer)# address 0.0.0.0 0.0.0.0
Device(config-ikev2-keyring-peer)# pre-shared-key cisco123
```

The following example shows how to configure an IKEv2 profile:

```
Device(config)# crypto ikev2 profile ikev2_prof
Device(config-ikev2-profile)# match identity remote address 120.1.1.2
Device(config-ikev2-profile)# authentication remote pre-share
Device(config-ikev2-profile)# authentication local pre-share
Device(config-ikev2-profile)# keyring local ikev2_key
Device(config-ikev2-profile)# dpd 10 2 periodic
end
```

The following example shows how to attach an IKEv2 profile to an IPSec profile:

```
Device(config)# crypto ipsec transform-set tfs esp-aes esp-sha-hmac
esn
Device(cfg-crypto-tran)# mode tunnel
end
Device(cfg-crypto-tran)# crypto ipsec profile ipsec_prof
Device(ipsec-profile)# set transform-set tfs
Device(ipsec-profile)# set ikev2-profile ikev2_prof
end
```

The following example shows how to create a tunnel interface and configure a tunnel source and tunnel destination under the tunnel interface:

```
Device(config)# interface Tunnel100
Device(config-if)# ip address 128.1.1.1 255.255.255.0
Device(config-if)# tunnel source 120.1.1.1
Device(config-if)# tunnel destination 120.1.1.2
Device(config-if)# tunnel protection ipsec profile ipsec_prof
end
```

Example: Configuring VRF aware GRE over IPsec

The following examples show how to configure a VRF aware GRE over IPsec tunnel.

The following example shows how to configure the FVRF instance and the IVRF instance:

```
Device# configure terminal
Device(config)# ip vrf fvrf
Device(config-vrf)# vrf definition CLIENT-VRF
Device(config-vrf)#address-family ipv4
Device(config-ipv4)# exit-address-family

Device# configure terminal
Device(config)# ip vrf ivrf
Device(config-vrf)# vrf definition WAN-VRF
Device(config-vrf)#address-family ipv4
Device(config-ipv4)# exit-address-family
```

The following example shows how to configure an Internet Key Exchange Version(IKEv2) key ring with symmetric preshared keys based on IP address:

```
Device# configure terminal
Device(config)# crypto ikev2 keyring ikev2_key
Device(config-ikev2-keyring)# peer mypeer
Device(config-ikev2-keyring-peer)# address 0.0.0.0 0.0.0.0
```

```
Device(config-ikev2-keyring-peer)# pre-shared-key cisco123
Device(config-ikev2-keyring-peer)# end
```

The following example shows how to configure an IKEv2 profile :

```
Device# configure terminal
Device(config)# crypto ikev2 profile ikev2_prof
Device(config-ikev2-profile)# match fvrf WAN-VRF
Device(config-ikev2-profile)#match identity remote address 130.1.1.1 255.255.255.255
Device(config-ikev2-profile)# authentication remote pre-share
Device(config-ikev2-profile)# authentication local pre-share
Device(config-ikev2-profile)# keyring local ikev2_key
Device(config-ikev2-profile)# dpd 10 2 periodic
Device(config-ikev2-profile)# end
```

The following example shows how to attach an IKEv2 profile to an IPSec profile:

```
Device# configure terminal
Device(config)# crypto ipsec transform-set tfs esp-aes esp-sha-hmac
 esn
Device(cfg-crypto-tran)# mode tunnel
Device(cfg-crypto-tran)# end

Device# configure terminal
Device(cfg-crypto-tran)# crypto ipsec profile ipsec_prof
Device(ipsec-profile)# set transform-set tfs
Device(ipsec-profile)# set ikev2-profile ikev2_prof
 responder-only
Device(ipsec-profile)# end
```

The following example shows how to configure a VRF aware GREoverIPSEC tunnel:

```
Device# configure terminal
Device(config)# interface Tunnel80
Device(config-if)# vrf forwarding CLIENT-VRF
Device(config-if)# ip address 172.16.1.1 255.255.255.0
Device(config-if)# tunnel source 130.1.1.2
Device(config-if)# tunnel destination 130.1.1.1
Device(config-if)# tunnel vrf WAN-VRF
Device(config-if)# tunnel protection ipsec profile ipsec_prof
Device(config-if)# end
```

Feature History for GRE over IPsec

This table provides release and related information for the features explained in this module.

These features are available on all releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Dublin 17.11.1	GRE over IPsec	The GRE over IPsec feature allows a payload to be GRE encapsulated and transferred securely over an IPsec tunnel.
Cisco IOS XE 17.13.1	VRF aware GRE over IPsec	VRF support was introduced for GRE over IPsec tunnels.

Use the Cisco Feature Navigator to find information about platform and software image support. To access the Cisco Feature Navigator, go to Cisco Feature Navigator.