LAN-to-LAN IPsec VPNs

A LAN-to-LAN VPN connects networks in different geographic locations.

You can create LAN-to-LAN IPsec connections with Cisco peers and with third-party peers that comply with all relevant standards. These peers can have any mix of inside and outside addresses using IPv4 and IPv6 addressing.

ASA does not allow locally sourced traffic other than ping to go over the VPN tunnel.

This chapter describes how to build a LAN-to-LAN VPN connection.

Summary of the Configuration

This section provides a summary of the example LAN-to-LAN configuration this chapter describes. Later sections provide step-by-step instructions.

```
hostname(config)# interface ethernet0/0
hostname(config-if)# ip address 10.10.4.100 255.255.0.0
hostname(config-if)# nameif outside
hostname(config-if)# no shutdown
hostname(config)# crypto ikev1 policy 1
hostname(config-ikev1-policy)# authentication pre-share
hostname(config-ikev1-policy)# encryption aes
hostname(config-ikev1-policy)# hash sha
hostname(config-ikev1-policy)# group 2
hostname(config-ikev1-policy)# lifetime 43200
hostname(config)# crypto ikev1 enable outside
hostname(config)# crypto ikev2 policy 1
hostname(config-ikev2-policy)# # encryption aes
hostname(config-ikev2-policy)# group 2
hostname(config-ikev12-policy)# prf sha
```

```
hostname(config-ikev2-policy)# lifetime 43200
hostname(config)# crypto ikev2 enable outside
hostname(config)#  crypto ipsec ikev1 transform-set FirstSet esp-aes esp-sha-hmac
hostname(config)# crypto ipsec ikev2 ipsec-proposal secure
hostname(config-ipsec-proposal)# protocol esp encryption aes
hostname(config-ipsec-proposal)# protocol esp integrity sha-1
hostname(config)# access-list l2l_list extended permit ip 192.168.0.0 255.255.0.0 150.150.0.0
 255.255.0.0
hostname(config)# tunnel-group 10.10.4.108 type ipsec-l2l
hostname(config)# tunnel-group 10.10.4.108 ipsec-attributes
hostname(config-tunnel-ipsec)# ikev1 pre-shared-key 44kkaol59636jnfx
hostname(config)# crypto map abcmap 1 match address l2l_list
hostname(config)# crypto map abcmap 1 set peer 10.10.4.108
hostname(config)# crypto map abcmap 1 set ikev1 transform-set FirstSet
hostname(config)# crypto map abcmap 1 set ikev2 ipsec-proposal secure
hostname(config)# crypto map abcmap interface outside
hostname(config)# write memory
```

Configure Site-to-Site VPN in Multi-Context Mode

Follow these steps to allow site-to-site support in multi-mode. By performing these steps, you can see how resource allocation breaks down.

Procedure

Step 1 To configure the VPN in multi-mode, configure a resource class and choose VPN licenses as part of the allowed resource. The "Configuring a Class for Resource Management" provides these configuration steps. The following is an example configuration:

```
class ctx1
 limit-resource VPN Burst Other 100
 limit-resource VPN Other 1000
```

Step 2 Configure a context and make it a member of the configured class that allows VPN licenses. The following is an example configuration:

```
context context1
 member ctx1
 allocate-interface GigabitEthernet3/0.2
 allocate-interface GigabitEthernet3/1.2
 allocate-interface Management0/0
 config-url disk0:/sm_s2s_ik1_ip4_no_webvpn.txt
 join-failover-group 1
```

Step 3 Configure connection profiles, policies, crypto maps, and so on, just as you would with single context VPN configuration of site-to-site VPN.

Configure Interfaces

An ASA has at least two interfaces, referred to here as outside and inside. Typically, the outside interface is connected to the public Internet, while the inside interface is connected to a private network and is protected from public access.

To begin, configure and enable two interfaces on the ASA. Then, assign a name, IP address and subnet mask. Optionally, configure its security level, speed, and duplex operation on the security appliance.

Note The ASA's outside interface address (for both IPv4/IPv6) cannot overlap with the private side address space.

Procedure

Step 1 To enter Interface configuration mode, in global configuration mode enter the **interface** command with the default name of the interface to configure. In the following example the interface is ethernet0.

```
hostname(config)# interface ethernet0/0
hostname(config-if)#
```

Step 2 To set the IP address and subnet mask for the interface, enter the **ip address** command. In the following example the IP address is 10.10.4.100 and the subnet mask is 255.255.0.0.

```
hostname(config-if)# ip address 10.10.4.100 255.255.0.0
hostname(config-if)#
```

Step 3 To name the interface, enter the **nameif** command, maximum of 48 characters. You cannot change this name after you set it. In the following example the name of the ethernet0 interface is outside.

```
hostname(config-if)# nameif outside
hostname(config-if)##
```

Step 4 To enable the interface, enter the **no** version of the **shutdown** command. By default, interfaces are disabled.

```
hostname(config-if)# no shutdown
hostname(config-if)#
```

Step 5 To save your changes, enter the **write memory** command:

```
hostname(config-if)# write memory
hostname(config-if)#
```

Step 6 To configure a second interface, use the same procedure.

Configure ISAKMP Policy and Enable ISAKMP on the Outside Interface

ISAKMP is the negotiation protocol that lets two hosts agree on how to build an IPsec security association (SA). It provides a common framework for agreeing on the format of SA attributes. This includes negotiating with the peer about the SA, and modifying or deleting the SA. ISAKMP separates negotiation into two phases: Phase 1 and Phase 2. Phase 1 creates the first tunnel, which protects later ISAKMP negotiation messages. Phase 2 creates the tunnel that protects data.

IKE uses ISAKMP to setup the SA for IPsec to use. IKE creates the cryptographic keys used to authenticate peers.

The ASA supports IKEv1 for connections from the legacy Cisco VPN client, and IKEv2 for the AnyConnect VPN client.

To set the terms of the ISAKMP negotiations, you create an IKE policy, which includes the following:

- The authentication type required of the IKEv1 peer, either RSA signature using certificates or preshared key (PSK).
- An encryption method, to protect the data and ensure privacy.
- A Hashed Message Authentication Codes (HMAC) method to ensure the identity of the sender, and to ensure that the message has not been modified in transit.
- A Diffie-Hellman group to determine the strength of the encryption-key-determination algorithm. The ASA uses this algorithm to derive the encryption and hash keys.
- For IKEv2, a separate pseudo-random function (PRF) used as the algorithm to derive keying material and hashing operations required for the IKEv2 tunnel encryption.
- A limit to the time the ASA uses an encryption key before replacing it.

With IKEv1 policies, for each parameter, you set one value. For IKEv2, you can configure multiple encryption and authentication types, and multiple integrity algorithms for a single policy. The ASA orders the settings from the most secure to the least secure and negotiates with the peer using that order. This allows you to potentially send a single proposal to convey all the allowed transforms instead of the need to send each allowed combination as with IKEv1.

The following sections provide procedures for creating IKEv1 and IKEv2 policies and enabling them on an interface:

- Configure ISAKMP Policies for IKEv1 Connections, on page 4
- Configure ISAKMP Policies for IKEv2 Connections, on page 6

Configure ISAKMP Policies for IKEv1 Connections

To configure ISAKMP policies for IKEv1 connections, use the **crypto ikev1 policy** priority command to enter IKEv1 policy configuration mode where you can configure the IKEv1 parameters.

Procedure

Step 1 Enter IPsec IKEv1 policy configuration mode. For example:

```
hostname(config)# crypto ikev1 policy 1
hostname(config-ikev1-policy)#
```

Step 2 Set the authentication method. The following example configures a preshared key:

```
hostname(config-ikev1-policy)# authentication pre-share
hostname(config-ikev1-policy)#
```

Step 3 Set the encryption method. The following example configures :

```
hostname(config-ikev1-policy)# encryption aes
hostname(config-ikev1-policy)#
```

Step 4 Set the HMAC method. The following example configures SHA-1:

```
hostname(config-ikev1-policy)# hash sha
hostname(config-ikev1-policy)#
```

Step 5 Set the Diffie-Hellman group. The following example configures Group 14:

```
hostname(config-ikev1-policy)# group 14
hostname(config-ikev1-policy)#
```

Step 6 Set the encryption key lifetime. The following example configures 43,200 seconds (12 hours):

```
hostname(config-ikev1-policy)# lifetime 43200
hostname(config-ikev1-policy)#
```

Step 7 Enable IKEv1 on the interface named outside in either single or multiple context mode:

```
hostname(config)# crypto ikev1 enable outside
hostname(config)#
```

Step 8 To save your changes, enter the **write memory** command:

```
hostname(config)# write memory
hostname(config)#
```

Configure ISAKMP Policies for IKEv2 Connections

To configure ISAKMP policies for IKEv2 connections, use the **crypto ikev2 policy** priority command to enter IKEv2 policy configuration mode where you can configure the IKEv2 parameters.

Procedure

Step 1 Enter IPsec IKEv2 policy configuration mode. For example:

```
hostname(config)# crypto ikev2 policy 1
hostname(config-ikev2-policy)#
```

Step 2 Set the encryption method. The following example configures AES :

```
hostname(config-ikev2-policy)# encryption aes
hostname(config-ikev2-policy)#
```

Step 3 Set the Diffie-Hellman group. The following example configures Group 15:

```
hostname(config-ikev2-policy)# group 15
hostname(config-ikev2-policy)#
```

Step 4 Set the pseudo-random function (PRF) used as the algorithm to derive keying material and hashing operations required for the IKEv2 tunnel encryption. The following example configures SHA-1 (an HMAC variant):

```
hostname(config-ikev12-policy)# prf sha
hostname(config-ikev2-policy)#
```

Step 5 Set the encryption key lifetime. The following example configures 43,200 seconds (12 hours):

```
hostname(config-ikev2-policy)# lifetime seconds 43200
hostname(config-ikev2-policy)#
```

Step 6 Enable IKEv2 on the interface named outside:

```
hostname(config)# crypto ikev2 enable outside
hostname(config)#
```

Step 7 To save your changes, enter the **write memory** command:

```
hostname(config)# write memory
hostname(config)#
```

Multiple Key Exchanges for IKEv2

IKEv2 uses Diffie-Hellman (DH) groups to establish a shared secret between an initiator and a responder. IKEv2 supports additional key exchanges to secure the IPsec communication from quantum computer attacks. Each exchange uses different DH groups. The computed shared secret for the SA setup is a combination of

all the keys derived from each exchange. An IKE SA is established after the multiple key exchanges between the IKE peers.

ASA uses seven new transform types for multiple key exchanges:

- Additional Key Exchange 1 (IANA value 6)
- Additional Key Exchange 2 (IANA value 7)
- Additional Key Exchange 3 (IANA value 8)
- Additional Key Exchange 4 (IANA value 9)
- Additional Key Exchange 5 (IANA value 10)
- Additional Key Exchange 6 (IANA value 11)
- Additional Key Exchange 7 (IANA value 12)

You can configure a maximum of seven multiple key exchanges. For each additional key exchange that you configure, you must specify the DH groups. ASA encrypts the intermediate key exchanges using the keys derived from the previous exchange. If the initiator and responder peers do not agree on a DH group, the negotiation fails and a **NO_PROPOSAL_CHOSEN** error notification is sent to the initiator. You can also configure the transform as none. If you choose none, the key exchange does not happen.

For an initiator, if the key exchange method is configured as **none** for an additional key exchange *n*:

- Responder can choose key exchange method as **none** for the additional key exchange *n*.
- Additional key exchange is optional.

For a successful proposal negotiation, all the transforms in the initiator's proposal must match with the transforms in the responder.

In the following example for an initiator:

```
crypto ikev2 policy 1
encryption aes
integrity sha256
group 14
prf sha256
lifetime seconds 120
additional-key-exchange 5
key-exchange-method none
```

Responder must have an additional-key-exchange 5 for the proposal to match.

If the peer does not support additional key exchange, one of the following occurs:

- If the initiator has another IKEv2 proposal that matches with the responder's proposal, an IKEv2 SA is established.
- The peer treats any additional key exchange transform type in the IKE_SA_INIT exchange message as unknown transform type and skips the proposals. The negotiation fails and a **NO_PROPOSAL_CHOSEN** error notification is sent to the initiator.

For more information about this feature, see RFC 9242.

Guidelines and Limitations for IKEv2 Multiple Key Exchanges

- You can have a maximum of seven multiple key exchanges.

- You cannot use the same DH group in the succeeding key exchanges.

For this feature, ASA does not support:

- IKEv1
- Combination of classical key exchange and post-quantum algorithm-based key exchange.
- Remote access VPN. Only site-to-site VPNs support IKEv2 multiple key exchanges.
- Clustering

Configure Multiple Key Exchanges for IKEv2

This configuration is optional, you can perform the configuration if you want to secure the IPsec communication from quantum computer attacks.

Before you begin

- Review the guidelines and limitations. For more information, see Guidelines and Limitations for IKEv2 Multiple Key Exchanges, on page 7.
- Configure the encryption algorithm, hash algorithm, authentication method, and SA lifetime for the IKEv2 policy. For more information, see Configure IKEv1 and IKEv2 Policies.

Procedure

Step 1 Create an IKEv2 policy.

crypto ikev2 policy *policy_index*

The prompt displays the IKEv2 policy configuration mode.

Example:

```
hostname(config)# crypto ikev2 policy 1
```

Step 2 Configure an additional key exchange transform for the IKEv2 policy.

additional-key-exchange <*1-7*>

The prompt displays the IKEv2 policy additional key exchange configuration mode. You can configure a maximum of seven key exchange transforms for a policy.

Example:

```
hostname(config-ikev2-policy)# additional-key-exchange 1
```

Step 3 Configure a key exchange method by defining one or more DH groups for the additional key exchange transform.

key-exchange-method <*DH_group*>

Specify the DH group as 14, 15, 16, 19, 20, 21, or 31. You can also configure the transform as none. If you choose none, the key exchange does not happen.

Example:

```
hostname(config-ikev2-policy-ake)# key-exchange-method 21 31
```

Step 4 Repeat steps 2 and 3 to configure multiple key exchanges for the IKEv2 policy.

```
hostname(config)# crypto ikev2 policy 1
hostname(config-ikev2-policy)# additional-key-exchange 1
hostname(config-ikev2-policy-ake)# key-exchange-method 21 31
hostname(config-ikev2-policy)# additional-key-exchange 2
hostname(config-ikev2-policy-ake)# key-exchange-method 20 21
hostname(config-ikev2-policy)# additional-key-exchange 3
hostname(config-ikev2-policy-ake)# key-exchange-method 19 20 none
...
```

What to do next

Verify the configuration. For more information, see Verify IKEv2 Multiple Key Exchanges Configurations, on page 9.

Verify IKEv2 Multiple Key Exchanges Configurations

Use the following show commands to view or verify the IKEv2 multiple key exchanges configurations:

- **show running-config crypto ikev2**

```
crypto ikev2 policy 1
encryption aes
integrity sha256
group 14
prf sha256
lifetime seconds 120
additional-key-exchange 1
key-exchange-method 21 31
additional-key-exchange 2
key-exchange-method 20 21
...
```

- **show crypto ikev2 sa detail**

```
IKEv2 SAs:
Session-id:4, Status:UP-ACTIVE, IKE count:1, CHILD count:1
Tunnel-id Local Remote fvrf/ivrf Status Role
41567725 192.168.15.1/500 192.168.15.2/500 READY INITIATOR
Encr: AES-CBC, keysize: 128, Hash: SHA96, DH Grp:14, Auth sign: PSK, Auth verify: PSK
Additional Key Exchange Group: AKE1: 31 AKE2: 21 AKE3: 20 AKE4: 19 AKE5: 16 AKE6: 15
AKE7: 14
Life/Active Time: 120/5 sec
Session-id: 4
Status Description: Negotiation done
Local spi: 6BB6B7BFA0BAADF4 Remote spi: 7030C7xxx xxxxxxE9DBDE77EB
Local id: 192.168.15.1
Remote id: 192.168.15.2
Local req mess id: 9 Remote req mess id: 0
Local next mess id: 9 Remote next mess id: 0
Local req queued: 9 Remote req queued: 0
Local window: 1 Remote window: 1
DPD configured for 10 seconds, retry 2
```

```
NAT-T is not detected
IKEv2 Fragmentation Configured MTU: 576 bytes, Overhead: 28 bytes, Effective MTU: 548
bytes
Parent SA Extended Status:
Delete in progress: FALSE
Marked for delete: FALSE
Child sa: local selector 20.0.0.0/0 - 20.0.0.255/65535
remote selector 30.0.0.0/0 - 30.0.0.255/65535
ESP spi in/out: 0x4a7d5da2/0x56a28fa8
AH spi in/out: 0x0/0x0
CPI in/out: 0x0/0x0
Encr: AES-CBC, keysize: 128, esp_hmac: SHA96
ah_hmac: None, comp: IPCOMP_NONE, mode tunnel
```

Create an IKEv1 Transform Set

An IKEv1 transform set combines an encryption method and an authentication method. During the IPsec security association negotiation with ISAKMP, the peers agree to use a particular transform set to protect a particular data flow. The transform set must be the same for both peers.

A transform set protects the data flows for the ACL specified in the associated crypto map entry. You can create transform sets in the ASA configuration, and then specify a maximum of 11 of them in a crypto map or dynamic crypto map entry.

The table below lists valid encryption and authentication methods.

Table 1: Valid Encryption and Authentication Methods

Valid Encryption Methods	Valid Authentication Methods
	esp-sha-hmac (default)
esp-aes (128-bit encryption) (default)	
esp-aes-192	
esp-aes-256	
esp-null	

Tunnel Mode is the usual way to implement IPsec between two ASAs that are connected over an untrusted network, such as the public Internet. Tunnel mode is the default and requires no configuration.

To configure a transform set, perform the following site-to-site tasks in either single or multiple context mode:

Procedure

Step 1 In global configuration mode enter the **crypto ipsec ikev1 transform-set** command. The following example configures a transform set with the name FirstSet, esp-aes encryption , and esp-sha-hmac authentication. The syntax is as follows:

esp-sha-hmac (default)

crypto ipsec ikev1 transform-set *transform-set-nameencryption-method authentication-method*

```
hostname(config)#
crypto ipsec transform-set FirstSet esp-aes esp-sha-hmac
hostname(config)#
```

Step 2 Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

Create an IKEv2 Proposal

For IKEv2, you can configure multiple encryption and authentication types, and multiple integrity algorithms for a single policy. The ASA orders the settings from the most secure to the least secure and negotiates with the peer using that order. This allows you to potentially send a single proposal to convey all the allowed transforms instead of the need to send each allowed combination as with IKEv1.

The table below lists valid IKEv2 encryption and authentication methods.

Table 2: Valid IKEv2 Encryption and Integrity Methods

Valid Encryption Methods	Valid Integrity Methods
	sha (default)
aes (default) - AES with a 128-bit key.	
aes-192	
aes-256	

To configure an IKEv2 proposal, perform the following tasks in either single or multiple context mode:

Procedure

Step 1 In global configuration mode, use the **crypto ipsec ikev2 ipsec-proposal** command to enter ipsec proposal configuration mode where you can specify multiple encryption and integrity types for the proposal. In this example, secure is the name of the proposal:

```
hostname(config)# crypto ipsec ikev2 ipsec-proposal secure
hostname(config-ipsec-proposal)#
```

Step 2 Then enter a protocol and encryption types. ESP is the only supported protocol. For example:

```
hostname(config-ipsec-proposal)# protocol esp encryption aes

hostname(config-ipsec-proposal)#
```

Step 3 Enter an integrity type. For example:

```
hostname(config-ipsec-proposal)# protocol esp integrity sha-1
hostname(config-ipsec-proposal)#
```

Step 4 Save your changes.

Configure an ACL

The ASA uses access control lists to control network access. By default, the adaptive security appliance denies all traffic. You need to configure an ACL that permits traffic. For more information, see "Information About Access Control Lists" in the general operations configuration guide.

The ACLs that you configure for this LAN-to-LAN VPN control connections are based on the source and translated destination IP addresses and, optionally, ports. Configure ACLs that mirror each other on both sides of the connection.

An ACL for VPN traffic uses the translated address.

Note For more information on configuring an ACL with a VPN filter, see the Specify a VLAN for Remote Access or Apply a Unified Access Control Rule to the Group Policy.

Procedure

Step 1 Enter the **access-list extended** command.

access-list *listname* **extended permit ip** *source-ipaddress source-netmask destination-ipaddress destination-netmask*

The following example configures an ACL named l2l_list that lets traffic from IP addresses in the 192.168.0.0 network travel to the 150.150.0.0 network.

```
hostname(config)# access-list l2l_list extended permit ip 192.168.0.0 255.255.0.0 150.150.0.0
 255.255.0.0
hostname(config)#
```

Step 2 Configure an ACL for the ASA on the other side of the connection that mirrors the ACL.

Subnets that are defined in an ACL in a crypto map, or in two different crypto ACLs that are attached to the same crypto map, should not overlap.

In the following example, the prompt for the peer is hostname2.

```
hostname2(config)# access-list l2l_list extended permit ip 150.150.0.0 255.255.0.0 192.168.0.0
 255.255.0.0
hostname(config)#
```

Define a Tunnel Group

A tunnel group is a set of records that contain tunnel connection policies. You configure a tunnel group to identify AAA servers, specify connection parameters, and define a default group policy. The ASA stores tunnel groups internally.

There are two default tunnel groups in the ASA: DefaultRAGroup, which is the default IPsec remote-access tunnel group, and DefaultL2Lgroup, which is the default IPsec LAN-to-LAN tunnel group. You can modify them, but not delete them.

The main difference between IKE versions 1 and 2 lies in terms of the authentication method they allow. IKEv1 allows only one type of authentication at both VPN ends (that is, either preshared key or certificate). However, IKEv2 allows asymmetric authentication methods to be configured (that is, preshared key authentication for the originator but certificate authentication for the responder) using separate local and remote authentication CLIs. Therefore, with IKEv2 you have asymmetric authentication, in which one side authenticates with one credential and the other side uses another credential (either a preshared key or certificate).

You can also create one or more new tunnel groups to suit your environment. The ASA uses these groups to configure default tunnel parameters for remote access and LAN-to-LAN tunnel groups when there is no specific tunnel group identified during tunnel negotiation.

To establish a basic LAN-to-LAN connection, you must set two attributes for a tunnel group:

- Set the connection type to IPsec LAN-to-LAN.
- Configure an authentication method for the IP address (that is, a preshared key for IKEv1 and IKEv2).

Procedure

Step 1 To set the connection type to IPsec LAN-to-LAN, enter the **tunnel-group** command.

The syntax is **tunnel-group** *name* **type** *type,* where name is the name you assign to the tunnel group, and type is the type of tunnel. The tunnel types as you enter them in the CLI are:

- **remote-access** (IPsec, SSL, and clientless SSL remote access)
- **ipsec-l2l** (IPsec LAN-to-LAN)

In the following example, the name of the tunnel group is the IP address of the LAN-to-LAN peer, 10.10.4.108.

```
hostname(config)# tunnel-group 10.10.4.108 type ipsec-l2l
hostname(config)#
```

Note LAN-to-LAN tunnel groups that have names that are not IP addresses can be used only if the tunnel authentication method is Digital Certificates and/or the peer is configured to use Aggressive Mode.

Step 2 To set the authentication method to use a preshared key, enter the ipsec-attributes mode and then enter the **ikev1pre-shared-key** command to create the preshared key. You need to use the same preshared key on both ASAs for this LAN-to-LAN connection.

The key is an alphanumeric string of 1-128 characters.

In the following example, the IKEv1 preshared key is 44kkaol59636jnfx:

```
hostname(config)# tunnel-group 10.10.4.108 ipsec-attributes
hostname(config-tunnel-ipsec)# ikev1-pre-shared-key 44kkaol59636jnfx
```

Step 3 Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

To verify that the tunnel is up and running, use the **show vpn-sessiondb summary**, **show vpn-sessiondb detail l2l**, or **show crypto ipsec sa** command.

Create a Crypto Map and Applying It To an Interface

Crypto map entries pull together the various elements of IPsec security associations, including the following:

- Which traffic IPsec should protect, which you define in an ACL.
- Where to send IPsec-protected traffic, by identifying the peer.
- What IPsec security applies to this traffic, which a transform set specifies.
- The local address for IPsec traffic, which you identify by applying the crypto map to an interface.

For IPsec to succeed, both peers must have crypto map entries with compatible configurations. For two crypto map entries to be compatible, they must, at a minimum, meet the following criteria:

- The crypto map entries must contain compatible crypto ACLs (for example, mirror image ACLs). If the responding peer uses dynamic crypto maps, the entries in the ASA crypto ACL must be “permitted” by the peer’s crypto ACL.
- The crypto map entries each must identify the other peer (unless the responding peer is using a dynamic crypto map).
- The crypto map entries must have at least one transform set in common.

If you create more than one crypto map entry for a given interface, use the sequence number (seq-num) of each entry to rank it: the lower the seq-num, the higher the priority. At the interface that has the crypto map set, the ASA evaluates traffic against the entries of higher priority maps first.

If Reverse Route Injection (RRI) is applied to a crypto map, that map must be unique to one interface on the ASA. In other words, the same crypto map cannot be applied to multiple interfaces. If more than one crypto map is applied to multiple interfaces, routes may not be cleaned up correctly. If multiple interfaces require a crypto map, each route must use a uniquely defined map.

Create multiple crypto map entries for a given interface if either of the following conditions exist:

- Different peers handle different data flows.
- You want to apply different IPsec security to different types of traffic (to the same or separate peers), for example, if you want traffic between one set of subnets to be authenticated, and traffic between

another set of subnets to be both authenticated and encrypted. In this case, define the different types of traffic in two separate ACLs, and create a separate crypto map entry for each crypto ACL.

Applying a Crypto Map on Multiple Interfaces

For a dual ISP, you can apply a crypto map to the external and backup interfaces on the ASA. The originate-only option is not available when you use this configuration. You must use Virtual Tunnel Interface (VTI) if you need this redundancy.

When you use a crypto map on multiple interfaces:

- You must have a routing protocol or route tracking.
- Ensure that the remote side also uses routing protocols.
- You must carefully choose multiple interfaces for the same crypto map as ASA allows a connection from a remote site on the interface with the less preferred route.

To create a crypto map and apply it to the outside interface in global configuration mode, perform the following steps in either single or multiple context mode:

Procedure

Step 1 To assign an ACL to a crypto map entry, enter the **crypto map match address** command.

The syntax is **crypto map** map-name seq-num **match address** aclname. In the following example the map name is abcmap, the sequence number is 1, and the ACL name is **l2l_list**.

```
hostname(config)# crypto map abcmap 1 match address l2l_list
hostname(config)#
```

Step 2 To identify the peer (s) for the IPsec connection, enter the **crypto map set peer** command.

The syntax is **crypto map** map-name seq-num **set peer** {ip_address1 | hostname1}[... ip_address10 | hostname10]. In the following example the peer name is 10.10.4.108.

```
hostname(config)# crypto map abcmap 1 set peer 10.10.4.108
hostname(config)#
```

Step 3 To specify an IKEv1 transform set for a crypto map entry, enter the **crypto map ikev1 set transform-set** command.

The syntax is **crypto map** map-name seq-num **ikev1 set transform-set** transform-set-name. In the following example, the transform set name is FirstSet.

```
hostname(config)# crypto map abcmap 1 set transform-set FirstSet
hostname(config)#
```

Step 4 To specify an IKEv2 proposal for a crypto map entry, enter the crypto map ikev2 set ipsec-proposal command:

The syntax is **crypto map** map-name seq-num set **ikev2 ipsec-proposal proposal-name**. In the following example, the proposal name is secure.

With the **crypto map** command, you can specify multiple IPsec proposals for a single map index. In that case, multiple proposals are transmitted to the IKEv2 peer as part of the negotiation, and the order of the proposals is determined by the administrator upon the ordering of the crypto map entry.

Note If combined mode (AES-GCM/GMAC) and normal mode (all others) algorithms exist in the IPsec proposal, then you cannot send a single proposal to the peer. You must have at least two proposals in this case, one for combined mode and one for normal mode algorithms.

```
hostname(config)# crypto map abcmap 1 set ikev2 ipsec-proposal secure
hostname(config)#
```

Apply Crypto Maps to Interfaces

You must apply a crypto map set to each interface through which IPsec traffic travels. The ASA supports IPsec on all interfaces. Applying the crypto map set to an interface instructs the ASA to evaluate all interface traffic against the crypto map set and to use the specified policy during connection or security association negotiations.

Binding a crypto map to an interface also initializes the runtime data structures, such as the security association database and the security policy database. When you later modify a crypto map in any way, the ASA automatically applies the changes to the running configuration. It drops any existing connections and reestablishes them after applying the new crypto map.

To apply the configured crypto map to the outside interface, perform the following steps:

Procedure

Step 1 Enter the **crypto map interface** command. The syntax is **crypto map** map-name **interface** interface-name.

```
hostname(config)# crypto map abcmap interface outside
hostname(config)#
```

Step 2 Save your changes.

```
hostname(config)# write memory
hostname(config)#
```