Configuring IPsec and ISAKMP

This chapter describes how to configure the IPsec and ISAKMP standards to build virtual private networks. It includes the following sections:

Tunneling Overview

Tunneling makes it possible to use a public TCP/IP network, such as the Internet, to create secure connections between remote users and a private corporate network. Each secure connection is called a tunnel.

The security appliance uses the ISAKMP and IPsec tunneling standards to build and manage tunnels. ISAKMP and IPsec accomplish the following:

- Negotiate tunnel parameters
- Establish tunnels
- Authenticate users and data
- Manage security keys
- Encrypt and decrypt data
- Manage data transfer across the tunnel
- Manage data transfer inbound and outbound as a tunnel endpoint or router

The security appliance functions as a bidirectional tunnel endpoint. It can receive plain packets from the private network, encapsulate them, create a tunnel, and send them to the other end of the tunnel where they are unencapsulated and sent to their final destination. It can also receive encapsulated packets from the public network, unencapsulate them, and send them to their final destination on the private network.

IPsec Overview

IPsec provides authentication and encryption services to prevent unauthorized viewing or modification of data within your network or as it travels over an unprotected network, such as the public Internet. Our implementation of the IPsec standard uses the ESP security protocol to provide authentication, encryption, and anti-replay services.

The security appliance uses IPsec for LAN-to-LAN VPN connections, and provides the option of using IPsec for client-to-LAN VPN connections. In IPsec terminology, a "peer" is a remote-access client or another secure gateway. For both connection types, the security appliance supports only Cisco peers. Because we adhere to VPN industry standards, ASAs may work with other vendors' peers; however, we do not support them.

During tunnel establishment, the two peers negotiate security associations that govern authentication, encryption, encapsulation, and key management. These negotiations involve two phases: first, to establish the tunnel (the IKE SA); and second, to govern traffic within the tunnel (the IPsec SA).

A LAN-to-LAN VPN connects networks in different geographic locations. In IPsec LAN-to-LAN connections, the security appliance can function as initiator or responder. In IPsec client-to-LAN connections, the security appliance functions only as responder. Initiators propose SAs; responders accept, reject, or make counter-proposals—all in accordance with configured SA parameters. To establish a connection, both entities must agree on the SAs.

Configuring ISAKMP

This section describes the Internet Key Exchange protocol which is also called the Internet Security Association and Key Management Protocol. The security appliance IKE commands use ISAKMP as a keyword, which this guide echoes. ISAKMP works with IPsec to make VPNs more scalable. This section includes the following topics:

ISAKMP Overview

IKE, also called ISAKMP, is the negotiation protocol that lets two hosts agree on how to build an IPsec security association. 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.

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

- An authentication method, to ensure the identity of the peers.

- 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 security appliance uses this algorithm to derive the encryption and hash keys.

- A limit for how long the security appliance uses an encryption key before replacing it.

Table 23-1 provides information about the ISAKMP policy keywords and their values.

Table 23-1   ISAKMP Policy Keywords for CLI Commands

Command	Keyword	Meaning	Description
isakmp policy authentication	rsa-sig	A digital certificate with keys generated by the RSA signatures algorithm	Specifies the authentication method the security appliance uses to establish the identity of each IPsec peer.
	dsa-sig	A digital certificate with keys generated by the DSA signatures algorithm	Specifies Digital Signature Algorithm signatures as the authentication method.
	pre-share (default)	Preshared keys	Preshared keys do not scale well with a growing network but are easier to set up in a small network.
isakmp policy encryption	des	56-bit DES-CBC	Specifies the symmetric encryption algorithm that protects data transmitted between two IPsec peers. The default is 168-bit Triple DES.
	3des (default)	168-bit Triple DES	
	aes aes-192 aes-256		The Advanced Encryption Standard supports key lengths of 128, 192, 256 bits.
isakmp policy hash	sha (default)	SHA-1 (HMAC variant)	Specifies the hash algorithm used to ensure data integrity. It ensures that a packet comes from where it says it comes from, and that it has not been modified in transit.
	md5	MD5 (HMAC variant)	The default is SHA-1. MD5 has a smaller digest and is considered to be slightly faster than SHA-1. A successful (but extremely difficult) attack against MD5 has occurred; however, the HMAC variant IKE uses prevents this attack.

Table 23-1   ISAKMP Policy Keywords for CLI Commands (continued)

Command	Keyword	Meaning	Description
isakmp policy group	1	Group 1 (768-bit)	Specifies the Diffie-Hellman group identifier, which the two IPsec peers use to derive a shared secret without transmitting it to each other.
	2 (default)	Group 2 (1024-bit)	
	5	Group 5 (1536-bit)	
	7	Group 7 (Elliptical curve field size is 163 bits.)	With the exception of Group 7, the lower the Diffie-Hellman group no., the less CPU time it requires to execute. The higher the Diffie-Hellman group no., the greater the security.
			Cisco VPN Client Version 3.x or higher requires a minimum of Group 2. (If you configure DH Group 1, the Cisco VPN Client cannot connect.)
			AES support is available on security appliances licensed for VPN-3DES only. To support the large key sizes required by AES, ISAKMP negotiation should use Diffie-Hellman (DH) Group 5.
			Designed for devices with low processing power, such as PDAs and mobile telephones, Group 7 provides the greatest security. The Certicom Movian Client requires Group 7.
isakmp policy lifetime	integer value (86400 = default)	120 to 2147483647 seconds	Specifies the SA lifetime. The default is 86,400 seconds or 24 hours. As a general rule, a shorter lifetime provides more secure ISAKMP negotiations (up to a point). However, with shorter lifetimes, the security appliance sets up future IPsec SAs more quickly.

Each configuration supports a maximum of 20 ISAKMP policies, each with a different set of values. Assign a unique priority to each policy you create. The lower the priority number, the higher the priority.

When ISAKMP negotiations begin, the peer that initiates the negotiation sends all of its policies to the remote peer, and the remote peer tries to find a match. The remote peer checks all of the peer's policies against each of its configured policies in priority order (highest priority first) until it discovers a match.

A match exists when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer policy specifies a lifetime less than or equal to the lifetime in the policy the initiator sent. If the lifetimes are not identical, the security appliance uses the shorter lifetime. If no acceptable match exists, ISAKMP refuses negotiation and the SA is not established.

There is an implicit trade-off between security and performance when you choose a specific value for each parameter. The level of security the default values provide is adequate for the security requirements of most organizations. If you are interoperating with a peer that supports only one of the values for a parameter, your choice is limited to that value.

Note   New ASA configurations do not have a default ISAKMP policy.

Configuring ISAKMP Policies

To configure ISAKMP policies, in global configuration mode, use the **isakmp policy** command with its various arguments. The syntax for ISAKMP policy commands is as follows:

isakmp policy *priority* **attribute_name** [**attribute_value** | *integer*]

You must include the priority in each of the ISAKMP commands. The priority number uniquely identifies the policy, and determines the priority of the policy in ISAKMP negotiations.

To enable and configure ISAKMP, complete the following steps, using the examples as a guide:

Note   If you do not specify a value for a given policy parameter, the default value applies.

Step 1   Specify the encryption algorithm. The default is Triple DES. This example sets encryption to DES.

```
isakmp policy priority encryption [aes | aes-192 | aes-256 | des | 3des]
```

For example:

```
hostname(config)# isakmp policy 2 encryption des
```

Step 2   Specify the hash algorithm. The default is SHA-1. This example configures MD5.

```
isakmp policy priority hash [md5 | sha]
```

For example:

```
hostname(config)# isakmp policy 2 hash md5
```

Step 3   Specify the authentication method. The default is preshared keys. This example configures RSA signatures.

```
isakmp policy priority authentication [pre-share | dsa-sig | rsa-sig]
```

For example:

```
hostname(config)# isakmp policy 2 authentication rsa-sig
```

Step 4   Specify the Diffie-Hellman group identifier. The default is Group 2. This example configures Group 5.

```
isakmp policy priority group [1 | 2 | 5 | 7]
```

For example:

```
hostname(config)# isakmp policy 2 group 5
```

Step 5   Specify the SA lifetime. This examples sets a lifetime of 4 hours (14400 seconds). The default is 86400 seconds (24 hours).

```
isakmp policy priority lifetime seconds
```

For example:

```
hostname(config)# isakmp policy 2 lifetime 14400
```

Enabling ISAKMP on the Outside Interface

You must enable ISAKMP on the interface that terminates the VPN tunnel. Typically this is the outside, or public interface.

To enable ISAKMP, enter the following command:

isakmp enable *interface-name*

For example:

```
hostname(config)# isakmp enable outside
```

Disabling ISAKMP in Aggressive Mode

Phase 1 ISAKMP negotiations can use either main mode or aggressive mode. Both provide the same services, but aggressive mode requires only two exchanges between the peers totaling 3 messages, rather than three exchanges totaling 6 messages. Aggressive mode is faster, but does not provide identity protection for the communicating parties. Therefore, the peers must exchange identification information prior to establishing a secure SA. Aggressive mode is enabled by default.

- Main mode is slower, using more exchanges, but it protects the identities of the communicating peers.

- Aggressive mode is faster, but does not protect the identities of the peers.

To disable ISAKMP in aggressive mode, enter the following command:

isakmp am-disable

For example:

```
hostname(config)# isakmp am-disable
```

If you have disabled aggressive mode, and want to revert to back to it, use the **no** form of the command. For example:

```
hostname(config)# no isakmp am-disable
```

Note   Disabling aggressive mode prevents Cisco VPN clients from using preshared key authentication to establish tunnels to the security appliance. However, they may use certificate-based authentication (that is, ASA or RSA) to establish tunnels.

Determining an ID Method for ISAKMP Peers

During Phase I ISAKMP negotiations the peers must identify themselves to each other. You can choose the identification method from the following options:

Address	Uses the IP addresses of the hosts exchanging ISAKMP identity information
Automatic	Determines ISAKMP negotiation by connection type: • IP address for preshared key • Cert Distinguished Name for certificate authentication

Hostname	Uses the fully qualified domain name of the hosts exchanging ISAKMP identity information (default). This name comprises the hostname and the domain name
Key ID	Uses the string the remote peer uses to look up the preshared key

The security appliance uses the Phase I ID to send to the peer. This is true for all VPN scenarios except LAN-to-LAN connections in main mode that authenticate with preshared keys.

The default setting is hostname.

To change the peer identification method, enter the following command:

isakmp identity {address | hostname | key-id *id-string* **| auto}**

For example, the following command sets the identification method to automatic:

```
hostname(config)# isakmp identity auto
```

Enabling IPsec over NAT-T

NAT-T lets IPsec peers establish a connection through a NAT device. It does this by encapsulating IPsec traffic in UDP datagrams, using port 4500, thereby providing NAT devices with port information. NAT-T auto-detects any NAT devices, and only encapsulates IPsec traffic when necessary. This feature is disabled by default.

- The security appliance can simultaneously support standard IPsec, IPsec over TCP, NAT-T, and IPsec over UDP, depending on the client with which it is exchanging data.
- When both NAT-T and IPsec over UDP are enabled, NAT-T takes precedence.
- When enabled, IPsec over TCP takes precedence over all other connection methods.
- When you enable NAT-T, the security appliance automatically opens port 4500 on all IPsec enabled interfaces.

The security appliance supports multiple IPsec peers behind a single NAT/PAT device operating in one of the following networks, but not both:

- LAN-to-LAN
- Remote access

In a mixed environment, the remote access tunnels fail the negotiation because all peers appear to be coming from the same public IP address, that of the NAT device. Also, remote access tunnels fail in a mixed environment because they often use the same name as the LAN-to-LAN tunnel group (that is, the IP address of the NAT device). This match can cause negotiation failures among multiple peers in a mixed LAN-to-LAN and remote access network of peers behind the NAT device.

Using NAT-T

To use NAT-T you must perform three tasks:

1. Enable IPsec over NAT-T globally on the security appliance.

2. Select the "before-fragmentation" option for the IPsec fragmentation policy. This option lets traffic travel across NAT devices that do not support IP fragmentation. It does not impede the operation of NAT devices that do support IP fragmentation.

3. Set a keepalive value, which can be from 10 to 3600 seconds. The default is 20 seconds.

To enable NAT-T globally on the security appliance, enter the following command:

isakmp nat-traversal *natkeepalive*

This example enables NAT-T and sets the keepalive to one hour.

hostname(config)# **isakmp nat-traversal 3600**

Enabling IPsec over TCP

IPsec over TCP enables a Cisco VPN client to operate in an environment in which standard ESP or ISAKMP cannot function, or can function only with modification to existing firewall rules. IPsec over TCP encapsulates both the ISAKMP and IPsec protocols within a TCP-like packet, and enables secure tunneling through both NAT and PAT devices and firewalls. This feature is disabled by default.

Note   This feature does not work with proxy-based firewalls.

IPsec over TCP works with remote access clients. You enable it globally, and it works on all ISAKMP enabled interfaces. It is a client to security appliance feature only. It does not work for LAN-to-LAN connections.

- The security appliance can simultaneously support standard IPsec, IPsec over TCP, NAT-Traversal, and IPsec over UDP, depending on the client with which it is exchanging data.
- The VPN 3002 hardware client, which supports one tunnel at a time, can connect using standard IPsec, IPsec over TCP, NAT-Traversal, or IPsec over UDP.
- When enabled, IPsec over TCP takes precedence over all other connection methods.

You enable IPsec over TCP on both the security appliance and the client to which it connects.

You can enable IPsec over TCP for up to 10 ports that you specify. If you enter a well-known port, for example port 80 (HTTP) or port 443 (HTTPS), the system displays a warning that the protocol associated with that port no longer works on the public interface. The consequence is that you can no longer use a browser to manage the security appliance through the public interface. To solve this problem, reconfigure the HTTP/HTTPS management to different ports.

The default port is 10000.

You must configure TCP port(s) on the client as well as on the security appliance. The client configuration must include at least one of the ports you set for the security appliance.

To enable IPsec over TCP globally on the security appliance, enter the following command:

isakmp ipsec-over-tcp [port *port 1...port0***]**

This example enables IPsec over TCP on port 45:

hostname(config)# **isakmp ctcp port 45**

Waiting for Active Sessions to Terminate Prior to Reboot

You can schedule a security appliance reboot to occur only when all active sessions have terminated voluntarily. This feature is disabled by default.

To enable waiting for all active sessions to voluntarily terminate before the security appliance reboots, enter the following command:

isakmp reload-wait

For example:

```
hostname(config)# isakmp reload-wait
```

Use the **reload** command to reboot the security appliance. If you set the **reload-wait** command, you can use the **reload quick** command to override the **reload-wait** setting. The **reload** and **reload-wait** commands are available in Privileged EXEC mode; neither includes the **isakmp** prefix.

Alerting Peers Before Disconnecting

Remote access or LAN-to-LAN sessions can drop for several reasons, such as: a security appliance shutdown or reboot, session idle timeout, maximum connection time exceeded, or administrator cut-off.

The security appliance can notify qualified peers (in LAN-to-LAN configurations), Cisco VPN Clients and VPN 3002 hardware clients of sessions that are about to be disconnected. The peer or client receiving the alert decodes the reason and displays it in the event log or in a pop-up panel. This feature is disabled by default.

Qualified clients and peers include the following:

- Security appliances with Alerts enabled.
- Cisco VPN clients running version 4.0 or later software (no configuration required).
- VPN 3002 hardware clients running version 4.0 or later software, and with Alerts enabled.
- VPN 3000 Series concentrators running version 4.0 or later software, with Alerts enabled.

To enable disconnect notification to IPsec peers, enter the **isakmp disconnect-notify command.**

For example:

```
hostname(config)# isakmp disconnect-notify
```

Configuring Certificate Group Matching

Tunnel groups define user connection terms and permissions. Certificate group matching lets you match a user to a tunnel group using either the Subject DN or Issuer DN of the user certificate.

To match users to tunnel groups based on these fields of the certificate, you must first create rules that define a matching criteria, and then associate each rule with the desired tunnel group.

To create a certificate map, use the **crypto ca certificate map** command. To define a tunnel group, use the **tunnel-group** command.

You must also configure a certificate group matching policy that sets one of the following methods for identifying the permission groups of certificate users:

- Match the group from the rules
- Match the group from the organizational unit (OU) field
- Use a default group for all certificate users

You can use any or all of these methods.

Creating a Certificate Group Matching Rule and Policy

To configure the policy and rules by which certificate-based ISAKMP sessions map to tunnel groups, and to associate the certificate map entries with tunnel groups, enter the **tunnel-group-map** command in global configuration mode.

The syntax follows:

tunnel-group-map enable {*rules | ou | ike-id | peer ip*}

tunnel-group-map [*rule-index*] **enable** *policy*

policy	Specifies the policy for deriving the tunnel group name from the certificate. *Policy* can be one of the following:
	ike-id—Indicates that if a tunnel-group is not determined based on a rule lookup or taken from the ou, then the certificate-based ISAKMP sessions are mapped to a tunnel group based on the content of the phase1 ISAKMP ID.
	ou—Indicates that if a tunnel-group is not determined based on a rule lookup, then use the value of the OU in the subject distinguished name (DN).
	peer-ip—Indicates that if a tunnel-group is not determined based on a rule lookup or taken from the ou or ike-id methods, then use the peer IP address.
	rules—Indicates that the certificate-based ISAKMP sessions are mapped to a tunnel group based on the certificate map associations configured by this command.
rule index	Optional. Refers to parameters specified by the **crypto ca certificate map** command. The values are 1 to 65535.

Be aware of the following:

- You can invoke this command multiple times as long as each invocation is unique and you do not reference a map index more than once.
- Rules cannot be longer than 255 characters.
- You can assign multiple rules to the same group. To do that, you add the rule priority and group first. Then you define as many criteria statements as you need for each group. When multiple rules are assigned to the same group, a match results for the first rule that tests true.
- Create a single rule if you want to require all criteria to match before assigning a user to a specific tunnel group. Requiring all criteria to match is equivalent to a logical AND operation. Alternatively, create one rule for each criterion if you want to require that only one match before assigning a user to a specific tunnel group. Requiring only one criterion to match is equivalent to a logical OR operation.

The following example enables mapping of certificate-based ISAKMP sessions to a tunnel group based on the content of the phase1 ISAKMP ID:

```
hostname(config)# tunnel-group-map enable ike-id
hostname(config)#
```

The following example enables mapping of certificate-based ISAKMP sessions to a tunnel group based on the IP address of the peer:

```
hostname(config)# tunnel-group-map enable peer-ip
hostname(config)#
```

The following example enables mapping of certificate-based ISAKMP sessions based on the organizational unit (OU) in the subject distinguished name (DN):

```
hostname(config)# tunnel-group-map enable ou
hostname(config)#
```

The following example enables mapping of certificate-based ISAKMP sessions based on established rules:

```
hostname(config)# tunnel-group-map enable rules
hostname(config)#
```

Using the Tunnel-group-map default-group Command

This command specifies a default tunnel group to use when the name cannot be derived by other configured methods.

The syntax is **tunnel-group-map** [*rule-index*] **default-group** *tunnel-group-name* where the *rule-index* is the priority for the rule, and *tunnel-group name* must be for a tunnel group that already exists.

Configuring IPsec

This section provides background information about IPsec and describes the procedures required to configure the security appliance when using IPsec to implement a VPN. It contains the following topics:

Understanding IPsec Tunnels

IPsec tunnels are sets of SAs that the security appliance establishes between peers. The SAs define the protocols and algorithms to apply to sensitive data, and also specify the keying material the peers use. IPsec SAs control the actual transmission of user traffic. SAs are unidirectional, but are generally established in pairs (inbound and outbound).

The peers negotiate the settings to use for each SA. Each SA consists of the following:

- Transform sets
- Crypto maps
- Access lists

- Tunnel groups
- Pre fragmentation policies

Understanding Transform Sets

A transform set is a combination of security protocols and algorithms that define how the security appliance protects data. You create multiple transform sets, and then specify up to six of them in a crypto map.

During IPsec SA negotiations, the peers must identify a transform set that is the same at both peers. The security appliance then applies the matching transform set to create an SA that protects data flows in the access list for that crypto map.

If you change a transform set definition, the security appliance tears down the tunnel. See "Clearing Security Associations" for further information.

Note If you clear or delete the only element in a transform set, the security appliance automatically removes the crypto map references to it.

Defining Crypto Maps

Crypto maps define the IPsec policy to be negotiated in the IPsec SA. They include the following:

- Access list to identify the packets that the IPsec connection permits and protects.
- Peer identification
- Local address for the IPsec traffic (See "Applying Crypto Maps to Interfaces" for more details.)
- Up to six transform sets with which to attempt to match the peer security settings.

A *crypto map set* consists of one or more crypto maps that have the same map name. You create a crypto map set when you create its first crypto map. The following command syntax creates or adds to a crypto map:

crypto map *map-name seq-num* **match address** *access-list-name*

You can continue to enter this command to add crypto maps to the crypto map set. In the following example, "mymap" is the name of the crypto map set to which you might want to add crypto maps:

crypto map mymap 10 match address 101

Among crypto maps with the same name, the *sequence number* (seq-num) shown in the syntax above distinguishes one from the other. The sequence number assigned to a crypto map also determines its priority among the other crypto maps within a crypto map set. The lower the sequence number, the higher the priority. After you assign a crypto map set to an interface, the security appliance evaluates all IP traffic passing through the interface against the crypto maps in the set, beginning with the crypto map with the lowest sequence number.

The *access control list* (ACL) assigned to a crypto map consists of all of the access control entries (ACEs) that have the same access-list-name, as shown in the following command syntax:

access-list *access-list-name* **{deny | permit} ip** *source source-netmask destination destination-netmask*

Each ACL consists of one or more ACEs that have the same access-list-name. You create an ACL when you create its first ACE. The following command syntax creates or adds to an ACL:

access-list *access-list-name* **{deny | permit} ip** *source source-netmask destination destination-netmask*

In the following example, the security appliance applies the IPsec protections assigned to the crypto map to all traffic flowing from the 10.0.0.0 subnet to the 10.1.1.0 subnet.

access-list 101 permit ip 10.0.0.0 255.255.255.0 10.1.1.0 255.255.255.0

The crypto map that matches the packet determines the security settings used in the SA negotiations. If the local security appliance initiates the negotiation, it uses the policy specified in the static crypto map to create the offer to send to the specified peer. If the peer initiates the negotiation, the security appliance attempts to match the policy to a static crypto map, and if that fails, any dynamic crypto maps in the crypto map set, to decide whether to accept or reject the peer offer.

For two peers to succeed in establishing an SA, they must have at least one compatible crypto map. To be compatible, a crypto map must meet the following criteria:

- The crypto map must contain compatible crypto ACLs (for example, mirror image ACLs). If the responding peer uses dynamic crypto maps, so must the security appliance as a requirement to apply IPsec.
- Each crypto map identifies the other peer (unless the responding peer uses dynamic crypto maps).
- The crypto maps have at least one transform set in common.

You can apply only one crypto map set to a single interface. Create multiple crypto maps for a particular interface on the security appliance if any of the following conditions exist:

- You want specific peers to handle different data flows.
- You want different IPsec security to apply to different types of traffic.

For example, create a crypto map and assign an ACL to identify traffic between two subnets and assign one transform set. Create another crypto map with a different ACL to identify traffic between another two subnets and apply a transform set with different VPN parameters.

If you create more than one crypto map for an interface, specify a sequence number (seq-num) for each map entry to determine its priority within the crypto map set.

Each ACE contains a permit or deny statement. Table 23-2 explains the special meanings of permit and deny ACEs in ACLs applied to crypto maps.

Table 23-2   Special Meanings of Permit and Deny in Crypto Access Lists Applied to Outbound Traffic

Result of Crypto Map Evaluation	Response
Match criterion in an ACE containing a permit statement	Halt further evaluation of the packet against the remaining ACEs in the crypto map set, and evaluate the packet security settings against those in the transform sets assigned to the crypto map. After matching the security settings to those in a transform set, the security appliance applies the associated IPsec settings. Typically for outbound traffic, this means that it decrypts, authenticates, and routes the packet.
Match criterion in an ACE containing a deny statement	Interrupt further evaluation of the packet against the remaining ACEs in the crypto map under evaluation, and resume evaluation against the ACEs in the next crypto map, as determined by the next seq-num assigned to it.
Fail to match all tested permit ACEs in the crypto map set	Route the packet without encrypting it.

ACEs containing deny statements filter out outbound traffic that does not require IPsec protection (for example, routing protocol traffic). Therefore, insert initial deny statements to filter outbound traffic that should not be evaluated against permit statements in a crypto access list.

For an inbound, encrypted packet, the security appliance uses the source address and ESP SPI to determine the decryption parameters. After the security appliance decrypts the packet, it compares the inner header of the decrypted packet to the permit ACEs in the ACL associated with the packet's SA. If the inner header fails to match the proxy, the security appliance drops the packet. It the inner header matches the proxy, the security appliance routes the packet.

When comparing the inner header of an inbound packet that was not encrypted, the security appliance ignores all deny rules because they would prevent the establishment of a Phase 2 SA.

Note   To route inbound, unencrypted traffic as clear text, insert deny ACEs before permit ACEs.

Figure 23-1 shows an example LAN-to-LAN network of security appliances.

Figure 23-1   Effect of Permit and Deny ACEs on Traffic (Conceptual Addresses)
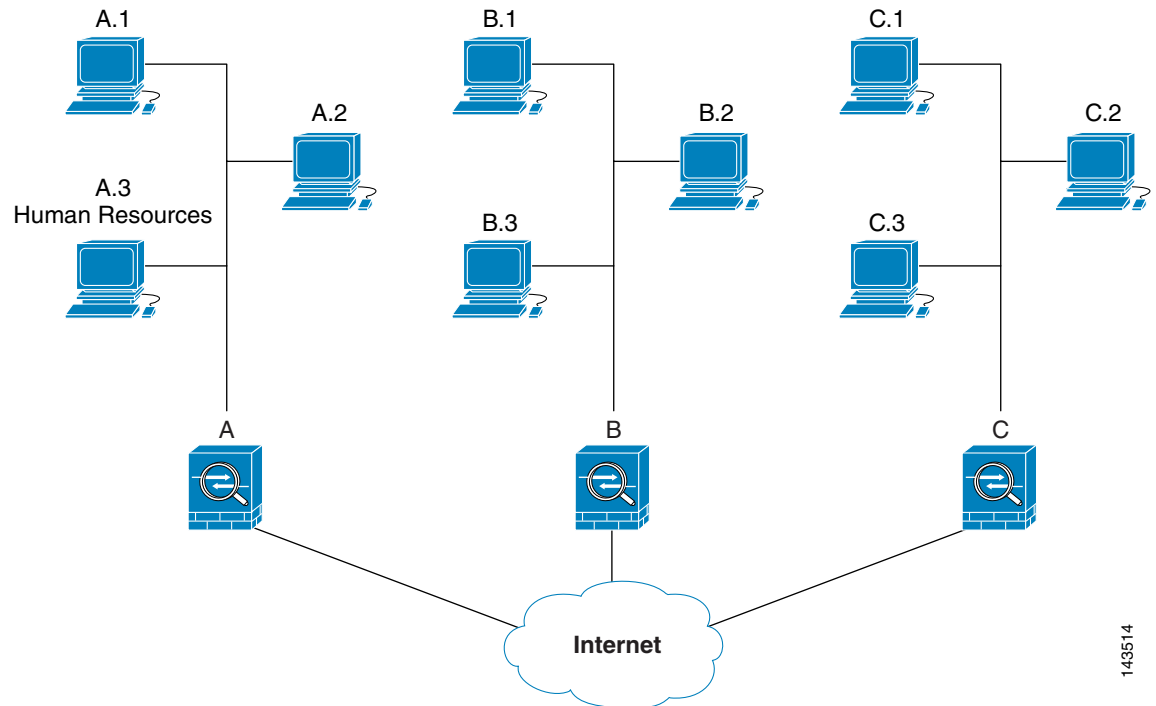
The simple address notation shown in this figure and used in the following explanation is an abstraction. An example with real IP addresses follows the explanation.

The objective in configuring Security Appliances A, B, and C in this example LAN-to-LAN network is to permit tunneling of all traffic originating from one of the hosts shown in Figure 23-1 and destined for one of the other hosts. However, because traffic from Host A.3 contains sensitive data from the Human Resources department, it requires strong encryption and more frequent rekeying than the other traffic. So we want to assign a special transform set for traffic from Host A.3.

To configure Security Appliance A for outbound traffic, we create two crypto maps, one for traffic from Host A.3 and the other for traffic from the other hosts in Network A, as shown in the following example:

```
Crypto Map Seq_No_1
    deny packets from A.3 to B
    deny packets from A.3 to C
    permit packets from A to B
    permit packets from A to C
Crypto Map Seq_No_2
    permit packets from A.3 to B
    permit packets from A.3 to C
```

After creating the ACLs, you assign a transform set to each crypto map to apply the required IPsec to each matching packet.

Cascading ACLs involves the insertion of deny ACEs to bypass evaluation against an ACL and resume evaluation against a subsequent ACL in the crypto map set. Because you can associate each crypto map with different IPsec settings, you can use deny ACEs to exclude special traffic from further evaluation in the corresponding crypto map, and match the special traffic to permit statements in another crypto map to provide or require different security. The sequence number assigned to the crypto ACL determines its position in the evaluation sequence within the crypto map set.

Figure 23-2 shows the cascading ACLs created from the conceptual ACEs above. Each symbol in the figure represents the following:

	Crypto map within a crypto map set.
	(Gap in a straight line) Exit from a crypto map when a packet matches an ACE.
	Packet that fits the description of one ACE. Each size ball represents a different packet matching the respective ACE in the figure. The differences in size merely represent differences in the source and destination of each packet.
	Redirection to the next crypto map in the crypto map set.
	Response when a packet either matches an ACE or fails to match all of the permit ACEs in a crypto map set.

Figure 23-2   Cascading ACLs in a Crypto Map Set

Crypto Map 1

Deny
A.3 B

Deny
A.3 C

Permit
A B

Permit
A C

Apply IPSec assigned to Crypto Map 1

Crypto Map 2

Permit
A.3 B

Permit
A.3 C

Apply IPSec
assigned to
Crypto Map 2

Route as clear text
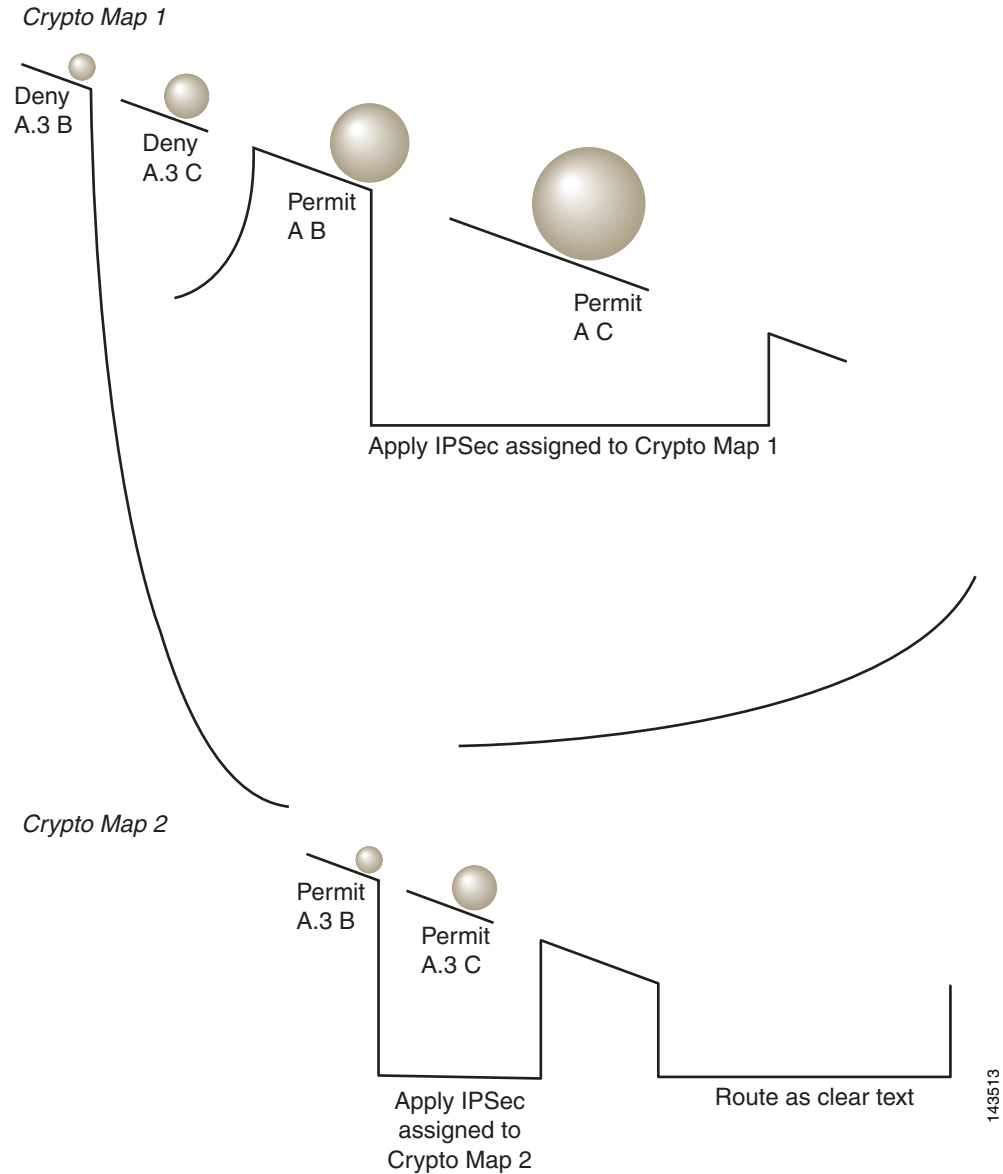
Security Appliance A evaluates a packet originating from Host A.3 until it matches a permit ACE and attempts to assign the IPsec security associated with the crypto map. Whenever the packet matches a deny ACE, the security appliance ignores the remaining ACEs in the crypto map and resumes evaluation against the next crypto map, as determined by the sequence number assigned to it. So in the example, if Security Appliance A receives a packet from Host A.3, it matches the packet to a deny ACE in the first crypto map and resumes evaluation of the packet against the next crypto map. When it matches the packet to the permit ACE in that crypto map, it applies the associated IPsec security (strong encryption and frequent rekeying).

To complete the security appliance configuration in the example network, we assign mirror crypto maps to Security Appliances B and C. However, because security appliances ignore deny ACEs when evaluating inbound, encrypted traffic, we can omit the mirror equivalents of the deny A.3 B and deny A.3 C ACEs, and therefore omit the mirror equivalents of Crypto Map 2. So the configuration of cascading ACLs in Security Appliances B and C is unnecessary.

Table 23-3 shows the ACLs assigned to the crypto maps configured for all three security appliances in Figure 23-1.

Table 23-3   Example Permit and Deny Statements (Conceptual)

Security Appliance A		Security Appliance B		Security Appliance C	
Crypto Map Sequence No.	ACE Pattern	Crypto Map Sequence No.	ACE Pattern	Crypto Map Sequence No.	ACE Pattern
1	deny A.3 B	1	permit B A	1	permit C A
	deny A.3 C				
	permit A B				
	permit A C		permit B C		permit C B
2	permit A.3 B				
	permit A.3 C				

Figure 23-3 maps the conceptual addresses shown in Figure 23-1 to real IP addresses.

Figure 23-3 *Effect of Permit and Deny ACEs on Traffic (Real Addresses)*
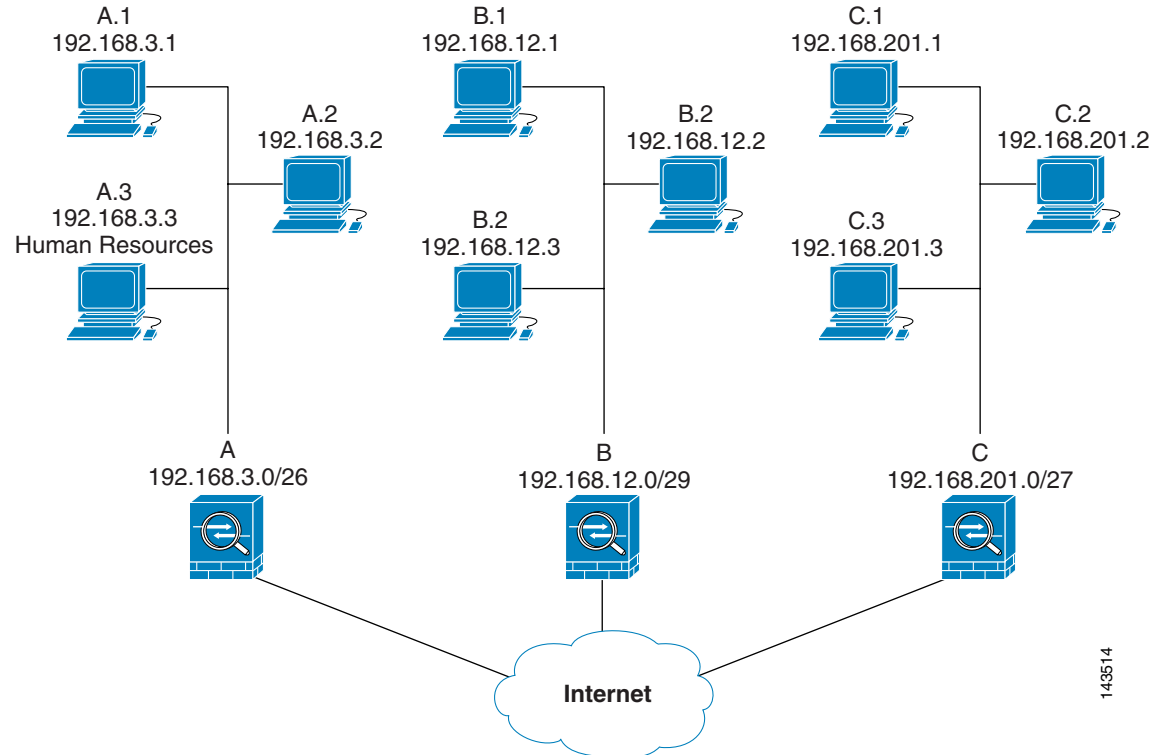
The tables that follow combine the IP addresses shown in Figure 23-3 to the concepts shown in Table 23-3. The real ACEs shown in these tables ensure that all IPsec packets under evaluation within this network receive the proper IPsec settings.

Table 23-4 *Example Permit and Deny Statements for Security Appliance A*

Security Appliance	Crypto Map Sequence No.	ACE Pattern	Real ACEs
A	1	deny A.3 B	deny 192.168.3.3 255.255.255.192 192.168.12.0 255.255.255.248
		deny A.3 C	deny 192.168.3.3 255.255.255.192 192.168.201.0 255.255.255.224
		permit A B	permit 192.168.3.0 255.255.255.192 192.168.12.0 255.255.255.248
		permit A C	permit 192.168.3.0 255.255.255.192 192.168.201.0 255.255.255.224
	2	permit A.3 B	permit 192.168.3.3 255.255.255.192 192.168.12.0 255.255.255.248
		permit A.3 C	permit 192.168.3.3 255.255.255.192 192.168.201.0 255.255.255.224
B	None needed	permit B A	permit 192.168.12.0 255.255.255.248 192.168.3.0 255.255.255.192
		permit B C	permit 192.168.12.0 255.255.255.248 192.168.201.0 255.255.255.224
C	None needed	permit C A	permit 192.168.201.0 255.255.255.224 192.168.3.0 255.255.255.192
		permit C B	permit 192.168.201.0 255.255.255.224 192.168.12.0 255.255.255.248

You can apply the same reasoning shown in the example network to use cascading ACLs to assign different security settings to different hosts or subnets protected by a Cisco security appliance.

Note   By default, the security appliance does not support IPsec traffic destined for the same interface from which it enters. (Names for this type of traffic include U-turn, hub-and-spoke, and hairpinning.) However, you might want IPsec to support U-turn traffic. To do so, insert an ACE to permit traffic to and from the network. For example, to support U-turn traffic on Security Appliance B, add a conceptual "permit B B" ACE to ACL1. The actual ACE would be as follows:

```
permit 192.168.12.0 255.255.255.248 192.168.12.0 255.255.255.248
```

Applying Crypto Maps to Interfaces

You must assign a crypto map set to each interface through which IPsec traffic flows. The security appliance supports IPsec on all interfaces. Assigning the crypto map set to an interface instructs the security appliance to evaluate all the traffic against the crypto map set and to use the specified policy during connection or SA negotiation.

Assigning a crypto map to an interface also initializes run-time data structures, such as the SA database and the security policy database. Reassigning a modified crypto map to the interface resynchronizes the run-time data structures with the crypto map configuration. Also, adding new peers through the use of new sequence numbers and reassigning the crypto map does not tear down existing connections.

Using Interface Access Lists

By default, the security appliance lets IPsec packets bypass interface ACLs. If you want to apply interface access lists to IPsec traffic, use the **no** form of the **sysopt connection permit-ipsec** command.

The crypto map access list bound to the outgoing interface either permits or denies IPsec packets through the VPN tunnel. IPsec authenticates and deciphers packets that arrive from an IPsec tunnel, and subjects them to evaluation against the ACL associated with the tunnel.

Access lists define which IP traffic to protect. For example, you can create access lists to protect all IP traffic between two subnets or two hosts. (These access lists are similar to access lists used with the **access-group** command. However, with the **access-group** command, the access list determines which traffic to forward or block at an interface.)

Before the assignment to crypto maps, the access lists are not specific to IPsec. Each crypto map references the access lists and determines the IPsec properties to apply to a packet if it matches a permit in one of the access lists.

Access lists assigned to IPsec crypto maps have four primary functions:

- Select outbound traffic to be protected by IPsec (permit = protect).

- Trigger an ISAKMP negotiation for data travelling without an established SA.

- Process inbound traffic to filter out and discard traffic that should have been protected by IPsec.

- Determine whether to accept requests for IPsec SAs when processing IKE negotiation from the peer. (Negotiation applies only to **ipsec-isakmp crypto map** entries.) The peer must "permit" a data flow associated with an **ipsec-isakmp crypto map** command entry to ensure acceptance during negotiation.

Regardless of whether the traffic is inbound or outbound, the security appliance evaluates traffic against the access lists assigned to an interface. You assign IPsec to an interface as follows:

Step 1   Create the access lists to be used for IPsec.

Step 2   Map the lists to one or more crypto maps, using the same crypto map name.

Step 3   Map the transform sets to the crypto maps to apply IPsec to the data flows.

Step 4   Apply the crypto maps collectively as a "crypto map set" by assigning the crypto map name they share to the interface.

In Figure 23-4, IPsec protection applies to traffic between Host 10.0.0.1 and Host 10.2.2.2 as the data exits the outside interface on Security Appliance A toward Host 10.2.2.2.

Figure 23-4   How Crypto Access Lists Apply to IPsec
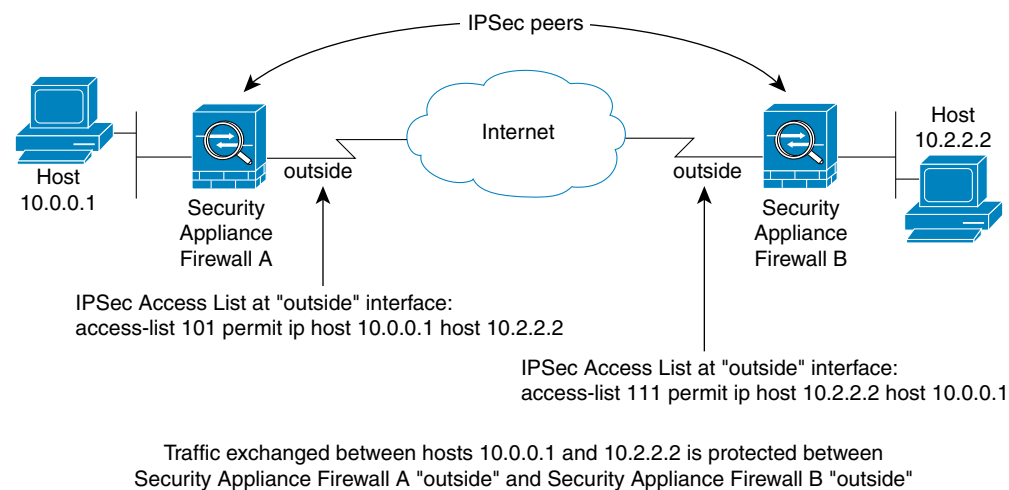
Security Appliance A evaluates traffic from Host 10.0.0.1 to Host 10.2.2.2, as follows:

- source = host 10.0.0.1
- dest = host 10.2.2.2

Security Appliance A also evaluates traffic from Host 10.2.2.2 to Host 10.0.0.1, as follows:

- source = host 10.2.2.2
- dest = host 10.0.0.1

The first permit statement that matches the packet under evaluation determines the scope of the IPsec SA.

Note   If you delete the only element in an access list, the security appliance also removes the associated crypto map.

If you modify an access list currently referenced by one or more crypto maps, use the **crypto map interface** command to re initialize the run-time SA database. See the **crypto map** command for more information.

We recommend that for every crypto access list specified for a static crypto map that you define at the local peer, you define a "mirror image" crypto access list at the remote peer. The crypto maps should also support common transforms and refer to the other system as a peer. This ensures correct processing of IPsec by both peers.

Note   Every static crypto map must define an access list and an IPsec peer. If either is missing, the crypto map is incomplete and the security appliance drops any traffic that it has not already matched to an earlier, complete crypto map. Use the **show conf** command to ensure that every crypto map is complete. To fix an incomplete crypto map, remove the crypto map, add the missing entries, and reapply it.

We discourage the use of the **any** keyword to specify source or destination addresses in crypto access lists because they cause problems. We strongly discourage the **permit any any** command statement because it does the following:

- Protects all outbound traffic, including all protected traffic sent to the peer specified in the corresponding crypto map.
- Requires protection for all inbound traffic.

  In this scenario, the security appliance silently drops all inbound packets that lack IPsec protection.

Be sure that you define which packets to protect. If you use the **any** keyword in a **permit** statement, preface it with a series of **deny** statements to filter out traffic that would otherwise fall within that **permit** statement that you do not want to protect.

Changing IPsec SA Lifetimes

You can change the global lifetime values that the security appliance uses when negotiating new IPsec SAs. You can override these global lifetime values for a particular crypto map.

IPsec SAs use a derived, shared, secret key. The key is an integral part of the SA; they time out together to require the key to refresh. Each SA has two lifetimes: "timed" and "traffic-volume." An SA expires after the respective lifetime and negotiations begin for a new one. The default lifetimes are 28,800 seconds (eight hours) and 4,608,000 kilobytes (10 megabytes per second for one hour).

If you change a global lifetime, the security appliance drops the tunnel. It uses the new value in the negotiation of subsequently established SAs.

When a crypto map does not have configured lifetime values and the security appliance requests a new SA, it inserts the global lifetime values used in the existing SA into the request sent to the peer. When a peer receives a negotiation request, it uses the smaller of either the lifetime value the peer proposes or the locally configured lifetime value as the lifetime of the new SA.

The peers negotiate a new SA before crossing the lifetime threshold of the existing SA to ensure that a new SA is ready when the existing one expires. The peers negotiate a new SA when about 5 to 15 percent of the lifetime of the existing SA remains.

Creating a Basic IPsec Configuration

The following steps cover basic IPsec configuration with static crypto maps.

Step 1   Create an access list to define the traffic to protect:

access-list *access-list-name* **{deny | permit} ip** *source source-netmask destination destination-netmask*

For example:

access-list 101 permit ip 10.0.0.0 255.255.255.0 10.1.1.0 255.255.255.0

In this example, the **permit** keyword causes all traffic that matches the specified conditions to be protected by crypto.

Step 2   Configure a transform set that defines how to protect the traffic. You can configure multiple transform sets, and then specify one or more of these transform sets in a crypto map (Step 3c).

crypto ipsec transform-set *transform-set-name transform1* [*tcansform2, transform3*]

For example:

crypto ipsec transform-set myset1 esp-des esp-sha-hmac
crypto ipsec transform-set myset2 esp-3des esp-sha-hmac
crypto ipsec transform-set aes_set esp-md5-hmac esp-aes-256

In this example, "myset1" and "myset2" and "aes_set" are the names of the transform sets.

Step 3   Create a crypto map by performing the following steps:

  a. Assign an access list to a crypto map:

crypto map *map-name seq-num* **match address** *access-list-name*

In the following example, "mymap" is the name of the crypto map set. The map set sequence number 10, which is used to rank multiple entries within one crypto map set. The lower the sequence number, the higher the priority.

crypto map mymap 10 match address 101

In this example, the access list named 101 is assigned to crypto map "mymap."

  b. Specify the peer to which the IPsec protected traffic can be forwarded:

crypto map *map-name seq-num* **set peer** *ip-address*

For example:

crypto map mymap 10 set peer 192.168.1.100

The security appliance sets ups an SA with the peer assigned the IP address 192.168.1.100. Specify multiple peers by repeating this command.

  c. Specify which transform sets are allowed for this crypto map. List multiple transform sets in order of priority (highest priority first). You can specify up to six transform sets.

crypto map *map-name seq-num* **set transform-set** *transform-set-name1* [*transform-set-name2, …transform-set-name6*]

For example:

crypto map mymap 10 set transform-set myset1 myset2

In this example, when traffic matches access list 101, the SA can use either "myset1" (first priority) or "myset2" (second priority) depending on which transform set matches the peer's transform set.

d. (Optional) Specify an SA lifetime for the crypto map if you want to override the global lifetime.

crypto map *map-name seq-num* **set security-association lifetime** {**seconds** seconds | **kilobytes** kilobytes}

For example:

crypto map mymap 10 set security-association lifetime seconds 2700
This example shortens the timed lifetime for the crypto map "mymap 10" to 2700 seconds (45 minutes). The traffic volume lifetime is not changed.

e. (Optional) Specify that IPsec require perfect forward secrecy when requesting new SA for this crypto map, or require PFS in requests received from the peer:

crypto map map-name *seq-num* **set pfs** [**group1** | **group2** | **group5** | **group7**]

For example:

crypto map mymap 10 set pfs group2

This example requires PFS when negotiating a new SA for the crypto map "mymap 10." The security appliance uses the 1024-bit Diffie-Hellman prime modulus group in the new SA.

Step 4   Apply a crypto map set to an interface for evaluating IPsec traffic:

crypto map *map-name* **interface** *interface-name*

For example:

crypto map mymap interface outside

In this example, the security appliance evaluates the traffic going through the outside interface against the crypto map "mymap" to determine whether it needs to be protected.

Using Dynamic Crypto Maps

Dynamic crypto maps can ease IPsec configuration and we recommend them for use in networks where the peers are not always predetermined. Use dynamic crypto maps for Cisco VPN clients (such as mobile users) and routers that obtain dynamically assigned IP addresses.

Note Use care when using the **any** keyword in **permit** entries in dynamic crypto maps. If the traffic covered by such a **permit** entry could include multicast or broadcast traffic, insert **deny** entries for the appropriate address range into the access list. Remember to insert **deny** entries for network and subnet broadcast traffic, and for any other traffic that IPsec should not protect.

Dynamic crypto maps work only to negotiate SAs with remote peers that initiate the connection. The security appliance cannot use dynamic crypto maps to initiate connections to a remote peer. With a dynamic crypto map, if outbound traffic matches a permit entry in an access list and the corresponding SA does not yet exist, the security appliance drops the traffic.

A dynamic crypto map is essentially a crypto map without all the parameters configured. It acts as a policy template where the missing parameters are later dynamically configured (as the result of an IPsec negotiation) to match the peer requirements. Dynamic crypto maps let peers exchange IPsec traffic with the security appliance even if the security appliance does not have a crypto map specifically configured that meets all the peer requirements.

Note   A dynamic crypto map requires only the **transform-set** parameter.

A crypto map set may include a dynamic crypto map. Dynamic crypto map sets should be the lowest priority crypto maps in the crypto map set (that is, they should have the highest sequence numbers) so that the security appliance evaluates other crypto maps first. It examines the dynamic crypto map set only when the other (static) map entries do not match.

Similar to static crypto map sets, a dynamic crypto map set consists of all of the dynamic crypto maps with the same dynamic-map-name. The dynamic-seq-num differentiates the dynamic crypto maps in a set. If you configure a dynamic crypto map, insert a permit ACL to identify the data flow of the IPsec peer for the crypto access list. Otherwise the security appliance accepts any data flow identity the peer proposes.

The procedure for using a crypto dynamic map entry is the same as the basic configuration described in "Creating a Basic IPsec Configuration," except that instead of creating a static crypto map, you create a crypto dynamic map entry. You can also combine static and dynamic map entries within a single crypto map set.

Create a crypto dynamic map entry by performing the following steps:

Step 1   (Optional) Assign an access list to a dynamic crypto map:

```
crypto dynamic-map dynamic-map-name dynamic-seq-num match address access-list-name
```

This determines which traffic should be protected and not protected.

For example:

```
crypto dynamic-map dyn1 10 match address 101
```

In this example, access list 101 is assigned to dynamic crypto map "dyn1." The map's sequence number is 10.

Step 2   Specify which transform sets are allowed for this dynamic crypto map. List multiple transform sets in order of priority (highest priority first).

```
crypto dynamic-map dynamic-map-name dynamic-seq-num set transform-set transform-set-name1,
[transform-set-name2, …transform-set-name9]
```

For example:

```
crypto dynamic-map dyn 10 set transform-set myset1 myset2
```

In this example, when traffic matches access list 101, the SA can use either "myset1" (first priority) or "myset2" (second priority), depending on which transform set matches the peer's transform sets.

Step 3   (Optional) Specify the SA lifetime for the crypto dynamic map entry if you want to override the global lifetime value:

```
crypto dynamic-map dynamic-map-name dynamic-seq-num set security-association lifetime
{seconds seconds | kilobytes kilobytes}
```

For example:

```
crypto dynamic-map dyn1 10 set security-association lifetime seconds 2700
```

This example shortens the timed lifetime for dynamic crypto map "dyn1 10" to 2700 seconds (45 minutes). The time volume lifetime is not changed.

Step 4 (Optional) Specify that IPsec ask for PFS when requesting new SAs for this dynamic crypto map, or should demand PFS in requests received from the peer:

```
crypto dynamic-map dynamic-map-name dynamic-seq-num set pfs [group1 | group2 | group5 |
group7]
```

For example:

```
crypto dynamic-map dyn1 10 set pfs group5
```

Step 5 Add the dynamic crypto map set into a static crypto map set.

Be sure to set the crypto maps referencing dynamic maps to be the lowest priority entries (highest sequence numbers) in a crypto map set.

```
crypto map map-name seq-num ipsec-isakmp dynamic dynamic-map-name
```

For example:

```
crypto map mymap 200 ipsec-isakmp dynamic dyn1
```

Providing Site-to-Site Redundancy

You can define multiple peers by using crypto maps to provide redundancy. This configuration is useful for site-to-site VPNs.

If one peer fails, the security appliance establishes a tunnel to the next peer associated with the crypto map. It sends data to the peer that it has successfully negotiated with, and that peer becomes the "active" peer. The "active" peer is the peer that the security appliance keeps trying first for follow-on negotiations until a negotiation fails. At that point the security appliance goes on to the next peer. The security appliance cycles back to the first peer when all peers associated with the crypto map have failed.

Viewing an IPsec Configuration

Table 23-5 lists commands you can enter to view information about your IPsec configuration.

Table 23-5   Commands to View IPsec Configuration Information

Command	Purpose
show running-configuration crypto	Displays the entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP.
show running-config crypto ipsec	Displays the complete IPsec configuration.
show running-config crypto isakmp	Displays the complete ISAKMP configuration.
show running-config crypto map	Displays the complete crypto map configuration.

Table 23-5   Commands to View IPsec Configuration Information (continued)

Command	Purpose
show running-config crypto dynamic-map	Displays the dynamic crypto map configuration.
show all crypto map	View all of the configuration parameters, including those with default values.

Clearing Security Associations

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take effect immediately, clear the existing SAs to reestablish them with the changed configuration. If the security appliance is actively processing IPsec traffic, it is desirable to clear only the portion of the SA database that the configuration changes would affect. Reserve clearing the full SA database for large-scale changes, or when the security appliance is processing a small amount of IPsec traffic.

Table 23-6 lists commands you can enter to clear and reinitialize IPsec SAs.

Table 23-6   Commands to Clear and Reinitialize IPsec SAs

Command	Purpose
clear configure crypto	Removes an entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP.
clear configure crypto ca trustpoint	Removes all trustpoints.
clear configure crypto dynamic-map	Removes all dynamic crypto maps. Includes keywords that let you remove specific dynamic crypto maps.
clear configure crypto map	Removes all crypto maps. Includes keywords that let you remove specific crypto maps.
clear configure isakmp	Removes the entire ISAKMP configuration.
clear configure isakmp policy	Removes all ISAKMP policies or a specific policy.
clear isakmp sa	Removes the entire ISAKMP SA database.

Clearing Crypto Map Configurations

The **clear configure crypto** command includes arguments that let you remove elements of the crypto configuration, including IPsec, crypto maps, dynamic crypto maps, CA trustpoints, all certificates, certificate map configurations, and ISAKMP.

Be aware that if you enter the **clear configure crypto** command without arguments, you remove the entire crypto configuration, including all certificates.

For more information, see the **clear configure crypto** command in the *Cisco Security Appliance Command Reference*.