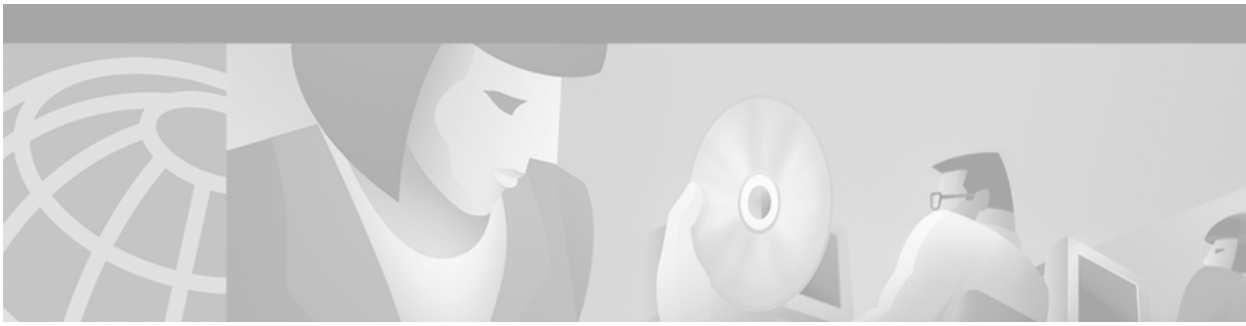


Configuring IPSec Network Security

This chapter describes how to configure IPSec, which is a framework of open standards developed by the Internet Engineering Task Force (IETF). IPSec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices ("peers"), such as Cisco routers.

IPSec provides the following network security services. These services are optional. In general, local security policy will dictate the use of one or more of these services:

- Data Confidentiality—The IPSec sender can encrypt packets before transmitting them across a network.
- Data Integrity—The IPSec receiver can authenticate packets sent by the IPSec sender to ensure that the data has not been altered during transmission.
- Data Origin Authentication—The IPSec receiver can authenticate the source of the IPSec packets sent. This service is dependent upon the data integrity service.
- Anti-Replay—The IPSec receiver can detect and reject replayed packets.

Note The term *data authentication* is generally used to mean data integrity and data origin authentication. Within this chapter it also includes anti-replay services, unless otherwise specified.

With IPSec, data can be transmitted across a public network without fear of observation, modification, or spoofing. This enables applications such as Virtual Private Networks (VPNs), including intranets, extranets, and remote user access.

For a complete description of the IPSec Network Security commands used in this chapter, refer to the "IPSec Network Security Commands" chapter in the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the chapter "Using Cisco IOS Software."

In This Chapter

This chapter has the following sections:

About IPSec

IPSec provides network data encryption at the IP packet level, offering a robust security solution that is standards-based. IPSec provides data authentication and anti-replay services in addition to data confidentiality services.

This section has the following sections:

Supported Standards

Cisco implements the following standards with this feature:

- IPSec—IP Security Protocol. IPSec is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. IPSec provides these security services at the IP layer; it uses IKE to handle negotiation of protocols and algorithms based on local policy, and to generate the encryption and authentication keys to be used by IPSec. IPSec can be used to protect one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host.

  Note The term IPSec is sometimes used to describe the entire protocol of IPSec data services and IKE security protocols and is also sometimes used to describe only the data services.

  IPSec is documented in a series of Internet Drafts, all available at http://www.ietf.org/html.charters/ipsec-charter.html. The overall IPSec implementation is per the latest version of the *Security Architecture for the Internet Protocol* Internet Draft (RFC2401). Cisco IOS IPSec implements RFC 2402 (*IP Authentication Header*) though RFC 2410 (*The NULL Encryption Algorithm and Its Use With IPSec*).

- Internet Key Exchange (IKE)—A hybrid protocol that implements Oakley and SKEME key exchanges inside the ISAKMP framework. While IKE can be used with other protocols, its initial implementation is with the IPSec protocol. IKE provides authentication of the IPSec peers, negotiates IPSec security associations, and establishes IPSec keys.

For more information on IKE, see the chapter "Configuring Internet Key Exchange Security Protocol."

The component technologies implemented for IPSec include:

- DES—The Data Encryption Standard (DES) is used to encrypt packet data. Cisco IOS implements the mandatory 56-bit DES-CBC with Explicit IV. Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet. For backwards compatibility, Cisco IOS IPSec also implements the RFC 1829 version of ESP DES-CBC.

  Cisco IOS also implements Triple DES (168-bit) encryption, depending on the software versions available for a specific platform. Triple DES (3DES) is a strong form of encryption that allows sensitive information to be transmitted over untrusted networks. It enables customers to utilize network layer encryption.

  Note Cisco IOS images with strong encryption (including, but not limited to, 56-bit data encryption feature sets) are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- MD5 (HMAC variant)—MD5 (Message Digest 5) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.
- SHA (HMAC variant)—SHA (Secure Hash Algorithm) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.

IPSec as implemented in Cisco IOS software supports the following additional standards:

- AH—Authentication Header. A security protocol which provides data authentication and optional anti-replay services. AH is embedded in the data to be protected (a full IP datagram).
- ESP—Encapsulating Security Payload. A security protocol which provides data privacy services and optional data authentication, and anti-replay services. ESP encapsulates the data to be protected.

List of Terms

Anti-Replay

Anti-replay is a security service where the receiver can reject old or duplicate packets in order to protect itself against replay attacks. IPSec provides this optional service by use of a sequence number combined with the use of data authentication. Cisco IOS IPSec provides this service whenever it provides the data authentication service, except in the following cases:

The service is not available for manually established security associations (that is, security associations established by configuration and not by IKE).

Data Authentication

Data Authentication includes two concepts:

- Data integrity (verify that data has not been altered).
- Data origin authentication (verify that the data was actually sent by the claimed sender).

Data authentication can refer either to integrity alone or to both of these concepts (although data origin authentication is dependent upon data integrity).

Data Confidentiality

Data confidentiality is a security service where the protected data cannot be observed.

Data Flow

Data flow is a grouping of traffic, identified by a combination of source address/mask, destination address/mask, IP next protocol field, and source and destination ports, where the protocol and port fields can have the values of **any**. In effect, all traffic matching a specific combination of these values is logically grouped together into a data flow. A data flow can represent a single TCP connection between two hosts, or it can represent all of the traffic between two subnets. IPSec protection is applied to data flows.

Peer

In the context of this chapter, "peer" refers to a router or other device that participates in IPSec.

Perfect Forward Secrecy (PFS)

Perfect forward secrecy is a cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.

Security Association

Security association is a description of how two or more entities will use security services in the context of a particular security protocol (AH or ESP) to communicate securely on behalf of a particular data flow. It includes such things as the transform and the shared secret keys to be used for protecting the traffic.

The IPSec security association is established either by IKE or by manual user configuration. Security associations are unidirectional and are unique per security protocol. So when security associations are established for IPSec, the security associations (for each protocol) for both directions are established at the same time.

When using IKE to establish the security associations for the data flow, the security associations are established when needed and expire after a period of time (or volume of traffic). If the security associations are manually established, they are established as soon as the necessary configuration is completed and do not expire.

Security Parameter Index (SPI)

Security parameter index is a number which, together with a destination IP address and security protocol, uniquely identifies a particular security association. When using IKE to establish the security associations, the SPI for each security association is a pseudo-randomly derived number. Without IKE, the SPI is manually specified for each security association.

Transform

Transform is the list of operations done on a dataflow to provide data authentication, data confidentiality, and data compression. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm; another transform is the AH protocol with the 56-bit DES encryption algorithm and the ESP protocol with the HMAC-SHA authentication algorithm.

Tunnel

In the context of this chapter, "tunnel" is a secure communication path between two peers, such as two routers. It does not refer to using IPSec in tunnel mode.

Supported Hardware, Switching Paths, and Encapsulation

IPSec has certain restrictions for hardware, switching paths, and encapsulation methods as follows:

Supported Hardware

ISA and ISM Support

For 7100 and 7200 hardware platforms, IPSec support requires the following adaptors or modules:

- Integrated Services Adapter (ISA) for the Cisco 7100 and 7200 series.
- Integrated Services Modules (ISM) for the Cisco 7100 series.

Note A VPN accelerator card and controller is also available on a Cisco 7100 and a Cisco 7200 series routers with an ISA and a Cisco 7100 series router with and ISM.

For more information on ISAs and ISMs, refer to the *Integrated Services Adapter and Integrated Services Module Installation and Configuration* publication.

VPN Module Support

For 1720 and 1750 hardware platforms, the VPN module assists the host processor by accelerating layer 3 IPSec data and voice encryption and decryption. The VPN module supports DES and 3DES encryption algorithms, MD5 and SHA-1 hashing, and Diffie-Hellman key generation.

The VPN module encrypts data using DES and Triple DES algorithms at speeds suitable for full duplex T1/E1 serial connections (4 megabits per second for 1518-byte packets). Equipped with a VPN module, a Cisco 1700 router supports up to 100 secure IPSec tunnels.

For more information on VPNs, refer to the chapter “Configuring Virtual Private Networks” in *Cisco IOS Dial Technologies Configuration Guide*.

AIMs and NM Support

For Cisco 2600 and Cisco 3600 series routers, the data encryption Advanced Integration Module (AIM) and Network Module (NM) provide hardware-based encryption. These data encryption products require Cisco IOS Release 12.1(3)XI or later.

The data encryption AIMs and NM are hardware Layer 3 (IPSec) encryption modules and provide DES and Triple DES IPSec encryption for multiple T1s or E1s of bandwidth. These products also have hardware support for Diffie-Hellman, RSA, and DSA key generation.

For more information on AIMs and NM, refer to *Installing the Data Encryption AIM in Cisco 2600 Series and Cisco 3600 Series Routers*.

Supported Switching Paths

Table 24 lists the supported switching paths that work with IPSec.

Table 24 Supported Switching Paths for IPSec

Switching Paths	Examples
Process switching	`interface ethernet0/0` ` no ip route-cache`

Fast switching	`interface ethernet0/0`
	` ip route-cache`
	`! Ensure that you will not hit flow switching.`
	` no ip route-cache flow`
	`! Disable CEF for the interface, which supercedes global CEF.`
	` no ip route-cache cef`

Cisco Express Forwarding (CEF)	`ip cef`
	`interface ethernet0/0`
	` ip route-cache`
	`! Ensure that you will not hit flow switching.`
	` no ip route-cache flow`

Fast-flow switching	`interface ethernet0/0`
	` ip route-cache`
	`! Enable flow switching`
	`  ip route-cache flow`
	`! Disable CEF for the interface.`
	`  no ip route-cache cef`

CEF-flow switching	`! Enable global CEF.`
	`ip cef`
	`interface ethernet0/0`
	` ip route-cache`
	` ip route-cache flow`
	`! Enable CEF for the interface`
	` ip route-cache cef`

For more information on the supported switching paths, see the *Cisco IOS Switching Services Configuration Guide*, Release 12.2.

Supported Encapsulation

IPSec works with the following serial encapsulations: High-Level Data-Links Control (HDLC), Point-to-Point Protocol (PPP), and Frame Relay.

IPSec also works with the GRE and IPinIP Layer 3, L2F, L2TP, DLSw+, and SRB tunneling protocols; however, multipoint tunnels are not supported. Other Layer 3 tunneling protocols may not be supported for use with IPSec.

Because the IPSec Working Group has not yet addressed the issue of group key distribution, IPSec currently cannot be used to protect group traffic (such as broadcast or multicast traffic).

Restrictions

At this time, IPSec can be applied to unicast IP datagrams only. Because the IPSec Working Group has not yet addressed the issue of group key distribution, IPSec does not currently work with multicasts or broadcast IP datagrams.

If you use Network Address Translation (NAT), you should configure static NAT translations so that IPSec will work properly. In general, NAT translation should occur before the router performs IPSec encapsulation; in other words, IPSec should be working with global addresses.

Overview of How IPSec Works

In simple terms, IPSec provides secure *tunnels* between two peers, such as two routers. You define which packets are considered sensitive and should be sent through these secure tunnels, and you define the parameters which should be used to protect these sensitive packets, by specifying characteristics of these tunnels. Then, when the IPSec peer sees such a sensitive packet, it sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer.

Note The use of the term *tunnel* in this chapter does not refer to using IPSec in tunnel mode.

More accurately, these *tunnels* are sets of security associations that are established between two IPSec peers. The security associations define which protocols and algorithms should be applied to sensitive packets, and also specify the keying material to be used by the two peers. Security associations are unidirectional and are established per security protocol (AH or ESP).

With IPSec you define what traffic should be protected between two IPSec peers by configuring access lists and applying these access lists to interfaces by way of crypto map sets. Therefore, traffic may be selected based on source and destination address, and optionally Layer 4 protocol, and port. (The access lists used for IPSec are used only to determine which traffic should be protected by IPSec, not which traffic should be blocked or permitted through the interface. Separate access lists define blocking and permitting at the interface.)

A crypto map set can contain multiple entries, each with a different access list. The crypto map entries are searched in order—the router attempts to match the packet to the access list specified in that entry.

When a packet matches a **permit** entry in a particular access list, and the corresponding crypto map entry is tagged as **cisco**, and connections are established if necessary. If the crypto map entry is tagged as **ipsec-isakmp**, IPSec is triggered. If no security association exists that IPSec can use to protect this traffic to the peer, IPSec uses IKE to negotiate with the remote peer to set up the necessary IPSec security associations on behalf of the data flow. The negotiation uses information specified in the crypto map entry as well as the data flow information from the specific access list entry. (The behavior is different for dynamic crypto map entries. Refer to the "Creating Dynamic Crypto Maps" section later in this chapter.)

If the crypto map entry is tagged as **ipsec-manual**, IPSec is triggered. If no security association exists that IPSec can use to protect this traffic to the peer, the traffic is dropped. In this case, the security associations are installed via the configuration, without the intervention of IKE. If the security associations did not exist, IPSec did not have all of the necessary pieces configured.

Once established, the set of security associations (outbound, to the peer) is then applied to the triggering packet as well as to subsequent applicable packets as those packets exit the router. "Applicable" packets are packets that match the same access list criteria that the original packet matched. For example, all applicable packets could be encrypted before being forwarded to the remote peer. The corresponding inbound security associations are used when processing the incoming traffic from that peer.

If IKE is used to establish the security associations, the security associations will have lifetimes so that they will periodically expire and require renegotiation. (This provides an additional level of security.)

Multiple IPSec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of security associations. For example, some data streams might be just authenticated while other data streams must both be encrypted and authenticated.

Access lists associated with IPSec crypto map entries also represent which traffic the router requires to be protected by IPSec. Inbound traffic is processed against the crypto map entries—if an unprotected packet matches a **permit** entry in a particular access list associated with an IPSec crypto map entry, that packet is dropped because it was not sent as an IPSec-protected packet.

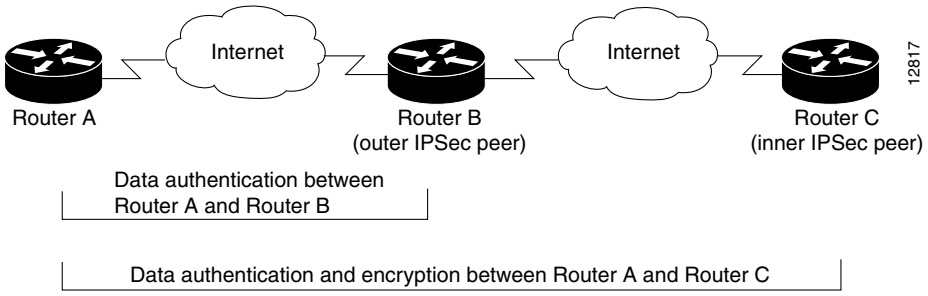
Crypto map entries also include transform sets. A transform set is an acceptable combination of security protocols, algorithms and other settings to apply to IPSec protected traffic. During the IPSec security association negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

Nesting of IPSec Traffic to Multiple Peers

You can nest IPSec traffic to a series of IPSec peers. For example, in order for traffic to traverse multiple firewalls (and these firewalls have a policy of not letting through traffic that they themselves have not authenticated), the router needs to establish IPSec tunnels with each firewall in turn. The "nearest" firewall becomes the "outermost" IPSec peer.

In the example shown in Figure 30, Router A encapsulates the traffic destined for Router C in IPSec (Router C is the IPSec peer). However, before Router A can send this traffic, it must first reencapsulate this traffic in IPSec in order to send it to Router B (Router B is the "outermost" IPSec peer).

Figure 30 Nesting Example of IPSec Peers



It is possible for the traffic between the "outer" peers to have one kind of protection (such as data authentication) and for traffic between the "inner" peers to have different protection (such as both data authentication and encryption).

Prerequisites

You must configure IKE as described in the "Configuring Internet Key Exchange Security Protocol" chapter.

Even if you decide to not use IKE, you still must disable it as described in the chapter "Configuring Internet Key Exchange Security Protocol."

IPSec Configuration Task List

- Ensuring That Access Lists Are Compatible with IPSec
- Setting Global Lifetimes for IPSec Security Associations

- Creating Crypto Access Lists
- Defining Transform Sets
- Creating Crypto Map Entries
- Applying Crypto Map Sets to Interfaces
- Monitoring and Maintaining IPSec

For IPSec configuration examples, refer to the "IPSec Configuration Example" section at the end of this chapter.

Ensuring That Access Lists Are Compatible with IPSec

IKE uses UDP port 500. The IPSec ESP and AH protocols use protocol numbers 50 and 51. Ensure that your access lists are configured so that protocol 50, 51, and UDP port 500 traffic is not blocked at interfaces used by IPSec. In some cases you might need to add a statement to your access lists to explicitly permit this traffic.

Setting Global Lifetimes for IPSec Security Associations

You can change the global lifetime values which are used when negotiating new IPSec security associations. (These global lifetime values can be overridden for a particular crypto map entry).

These lifetimes only apply to security associations established via IKE. Manually established security associations do not expire.

There are two lifetimes: a "timed" lifetime and a "traffic-volume" lifetime. A security association expires after the first of these lifetimes is reached. The default lifetimes are 3600 seconds (one hour) and 4,608,000 kilobytes (10 megabits per second for one hour per second for one hour).

If you change a global lifetime, the new lifetime value will not be applied to currently existing security associations, but will be used in the negotiation of subsequently established security associations. If you wish to use the new values immediately, you can clear all or part of the security association database. Refer to the **clear crypto sa** command for more details.

IPSec security associations use one or more shared secret keys. These keys and their security associations time out together.

To change a global lifetime for IPSec security associations, use one or more of the following commands in global configuration mode:

Command	Purpose
`Router(config)#` **`crypto ipsec security-association lifetime seconds`** *`seconds`*	Changes the global "timed" lifetime for IPSec SAs. This command causes the security association to time out after the specified number of seconds have passed.

Command	Purpose
`Router(config)# crypto ipsec security-association lifetime kilobytes` *kilobytes*	Changes the global "traffic-volume" lifetime for IPSec SAs. This command causes the security association to time out after the specified amount of traffic (in kilobytes) have passed through the IPSec "tunnel" using the security association.
`Router(config)# clear crypto sa` or `Router(config)# clear crypto sa peer` {*ip-address* | *peer-name*} or `Router(config)# clear crypto sa map` *map-name* or `Router (config)# clear crypto sa entry` *destination-address protocol spi*	(Optional) Clears existing security associations. This causes any existing security associations to expire immediately; future security associations will use the new lifetimes. Otherwise, any existing security associations will expire according to the previously configured lifetimes. **Note** Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, see the **clear crypto sa** command.

How These Lifetimes Work

Assuming that the particular crypto map entry does not have lifetime values configured, when the router requests new security associations it will specify its global lifetime values in the request to the peer; it will use this value as the lifetime of the new security associations. When the router receives a negotiation request from the peer, it will use the smaller of either the lifetime value proposed by the peer or the locally configured lifetime value as the lifetime of the new security associations.

The security association (and corresponding keys) will expire according to whichever comes sooner, either after the number of seconds has passed (specified by the **seconds** keyword) or after the amount of traffic in kilobytes is passed (specified by the **kilobytes** keyword). Security associations that are established manually (via a crypto map entry marked as **ipsec-manual**) have an infinite lifetime.

A new security association is negotiated *before* the lifetime threshold of the existing security association is reached, to ensure that a new security association is ready for use when the old one expires. The new security association is negotiated either 30 seconds before the **seconds** lifetime expires or when the volume of traffic through the tunnel reaches 256 kilobytes less than the **kilobytes** lifetime (whichever comes first).

If no traffic has passed through the tunnel during the entire life of the security association, a new security association is not negotiated when the lifetime expires. Instead, a new security association will be negotiated only when IPSec sees another packet that should be protected.

Creating Crypto Access Lists

Crypto access lists are used to define which IP traffic will be protected by crypto and which traffic will not be protected by crypto. (These access lists are *not* the same as regular access lists, which determine what traffic to forward or block at an interface.) For example, access lists can be created to protect all IP traffic between Subnet A and Subnet Y or Telnet traffic between Host A and Host B.

The access lists themselves are not specific to IPSec. It is the crypto map entry referencing the specific access list that defines whether IPSec processing is applied to the traffic matching a **permit** in the access list.

Crypto access lists associated with IPSec crypto map entries have four primary functions:

- Select outbound traffic to be protected by IPSec (permit = protect).
- Indicate the data flow to be protected by the new security associations (specified by a single **permit** entry) when initiating negotiations for IPSec security associations.
- Process inbound traffic in order to filter out and discard traffic that should have been protected by IPSec.
- Determine whether or not to accept requests for IPSec security associations on behalf of the requested data flows when processing IKE negotiation from the IPSec peer. (Negotiation is only done for **ipsec-isakmp** crypto map entries.) In order to be accepted, if the peer initiates the IPSec negotiation, it must specify a data flow that is "permitted" by a crypto access list associated with an **ipsec-isakmp** crypto map entry.

If you want certain traffic to receive one combination of IPSec protection (for example, authentication only) and other traffic to receive a different combination of IPSec protection (for example, both authentication and encryption), you need to create two different crypto access lists to define the two different types of traffic. These different access lists are then used in different crypto map entries which specify different IPSec policies.

Later, you will associate the crypto access lists to particular interfaces when you configure and apply crypto map sets to the interfaces (following instructions in the sections "Creating Crypto Map Entries" and "Applying Crypto Map Sets to Interfaces").

To create crypto access lists, use the following command in global configuration mode:

Command	Purpose
`Router(config)# access-list` *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**] or `Router(config)# ip access-list extended` *name* Follow with **permit** and **deny** statements as appropriate.	Specifies conditions to determine which IP packets will be protected.[1] (Enable or disable crypto for traffic that matches these conditions.) Configure "mirror image" crypto access lists for use by IPSec and avoid using the **any** keyword, as described in the sections "Defining Mirrror Image Crypto Access Lists at Each IPSec Peer" and "Using the any Keyword in Crypto Access Lists" (following). Also see the "Crypto Access List Tips" section.

1. You specify conditions using an IP access list designated by either a number or a name. The **access-list** command designates a numbered extended access list; the **ip access-list extended** command designates a named access list.

This section contains the following sections:

- Crypto Access List Tips
- Defining Mirrror Image Crypto Access Lists at Each IPSec Peer
- Using the any Keyword in Crypto Access Lists

Crypto Access List Tips

Using the **permit** keyword causes all IP traffic that matches the specified conditions to be protected by crypto, using the policy described by the corresponding crypto map entry. Using the **deny** keyword prevents traffic from being protected by crypto in the context of that particular crypto map entry. (In other words, it does not allow the policy as specified in this crypto map entry to be applied to this traffic.) If this traffic is denied in all of the crypto map entries for that interface, then the traffic is not protected by crypto.

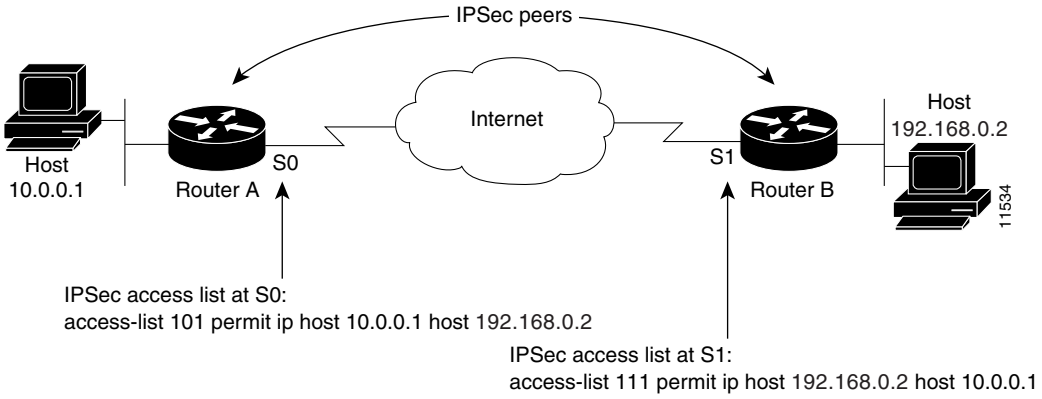
The crypto access list you define will be applied to an interface after you define the corresponding crypto map entry and apply the crypto map set to the interface. Different access lists must be used in different entries of the same crypto map set. (These two tasks are described in following sections.) However, both inbound and outbound traffic will be evaluated against the same "outbound" IPSec access list. Therefore, the access list's criteria is applied in the forward direction to traffic exiting your router, and the reverse direction to traffic entering your router. In Figure 31, IPSec protection is applied to traffic between Host 10.0.0.1 and Host 20.0.0.2 as the data exits Router A's S0 interface en route to Host 20.0.0.2. For traffic from Host 10.0.0.1 to Host 20.0.0.2, the access list entry on Router A is evaluated as follows:

```
source = host 10.0.0.1
dest = host 20.0.0.2
```

For traffic from Host 20.0.0.2 to Host 10.0.0.1, that same access list entry on Router A is evaluated as follows:

```
source = host 20.0.0.2
dest = host 10.0.0.1
```

Figure 31 How Crypto Access Lists Are Applied for Processing IPSec



If you configure multiple statements for a given crypto access list which is used for IPSec, in general the first **permit** statement that is matched will be the statement used to determine the scope of the IPSec security association. That is, the IPSec security association will be set up to protect traffic that meets the criteria of the matched statement only. Later, if traffic matches a different **permit** statement of the crypto access list, a new, separate IPSec security association will be negotiated to protect traffic matching the newly matched access list statement.

Note Access lists for crypto map entries tagged as **ipsec-manual** are restricted to a single **permit** entry and subsequent entries are ignored. In other words, the security associations established by that particular crypto map entry are only for a single data flow. To be able to support multiple manually established security associations for different kinds of traffic, define multiple crypto access lists, and then apply each one to a separate **ipsec-manual** crypto map entry. Each access list should include one **permit** statement defining what traffic to protect.

Any unprotected inbound traffic that matches a **permit** entry in the crypto access list for a crypto map entry flagged as IPSec will be dropped, because this traffic was expected to be protected by IPSec.

Note If you view your router's access lists by using a command such as **show ip access-lists**, *all* extended IP access lists will be shown in the command output. This includes extended IP access lists that are used for traffic filtering purposes as well as those that are used for crypto. The **show** command output does not differentiate between the different uses of the extended access lists.

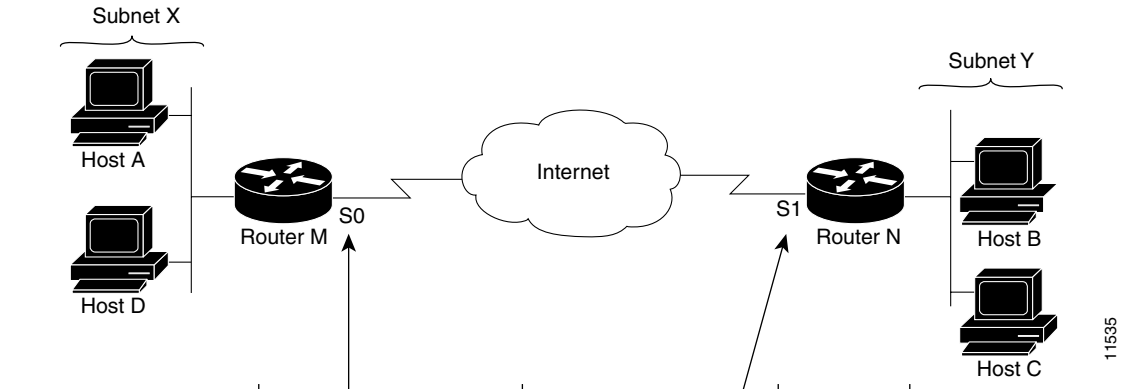
See the *Cisco IOS Security Command Reference* for complete details about the extended IP access list commands used to create IPSec access lists.

Defining Mirror Image Crypto Access Lists at Each IPSec Peer

For every crypto access list specified for a static crypto map entry that you define at the local peer, you must define a "mirror image" crypto access list at the remote peer. This ensures that traffic that has IPSec protection applied locally can be processed correctly at the remote peer. (The crypto map entries themselves must also support common transforms and must refer to the other system as a peer.)

Figure 32 shows some sample scenarios when you have mirror image access lists and when you do not have mirror image access lists.

Figure 32 Mirror Image vs. Non-Mirror Image Crypto Access Lists (for IPSec)



		IPSec access list at S0:	IPSec access list at S1:	1st packet	Result
Mirror image access lists at Router M S0 and Router N S1	Case 1	permits Host A → Host B	permits Host B → Host A	A → B or B → A	SAs established for traffic A↔B (good)
	Case 2	permits Subnet X → Subnet Y	permits Subnet Y → Subnet X	A → B or B → A or A → C or C → D and so on	SAs established for traffic X↔Y (good)
	Case 3	permits Host A → Host B	permits Subnet Y → Subnet X	A → B	SAs established for traffic A↔B (good)
	Case 4			B → A	SAs cannot be established and packets from Host B to Host A are dropped (bad)

As Figure 32 indicates, IPSec Security Associations (SAs) can be established as expected whenever the two peers' crypto access lists are mirror images of each other. However, an IPSec SA can be established only some of the time when the access lists are not mirror images of each other. This can happen in the case where an entry in one peer's access list is a subset of an entry in the other peer's access list, such

as shown in Cases 3 and 4 of Figure 32. IPSec SA establishment is critical to IPSec—without SAs, IPSec does not work, causing any packets matching the crypto access list criteria to be silently dropped instead of being forwarded with IPSec security.

In Figure 32, an SA cannot be established in Case 4. This is because SAs are always requested according to the crypto access lists at the initiating packet's end. In Case 4, Router N requests that all traffic between Subnet X and Subnet Y be protected, but this is a superset of the specific flows permitted by the crypto access list at Router M so the request is therefore not permitted. Case 3 works because Router M's request is a subset of the specific flows permitted by the crypto access list at Router N.

Because of the complexities introduced when crypto access lists are not configured as mirror images at peer IPSec devices, Cisco strongly encourages you to use mirror image crypto access lists.

Using the any Keyword in Crypto Access Lists

When you create crypto access lists, using the **any** keyword could cause problems. Cisco discourages the use of the **any** keyword to specify source or destination addresses.

The **any** keyword in a **permit** statement is discouraged when you have multicast traffic flowing through the IPSec interface; the **any** keyword can cause multicast traffic to fail.

The **permit any any** statement is strongly discouraged, as this will cause all outbound traffic to be protected (and all protected traffic sent to the peer specified in the corresponding crypto map entry) and will require protection for all inbound traffic. Then, all inbound packets that lack IPSec protection will be silently dropped, including packets for routing protocols, NTP, echo, echo response, and so on.

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

Defining Transform Sets

A transform set represents a certain combination of security protocols and algorithms. During the IPSec security association negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

You can specify multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. The transform set defined in the crypto map entry will be used in the IPSec security association negotiation to protect the data flows specified by that crypto map entry's access list.

During IPSec security association negotiations with IKE, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and will be applied to the protected traffic as part of both peers' IPSec security associations.

With manually established security associations, there is no negotiation with the peer, so both sides must specify the same transform set.

If you change a transform set definition, the change is only applied to crypto map entries that reference the transform set. The change will not be applied to existing security associations, but will be used in subsequent negotiations to establish new security associations. If you want the new settings to take effect sooner, you can clear all or part of the security association database by using the **clear crypto sa** command.

To define a transform set, use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto ipsec transform-set transform-set-name transform1 [transform2 [transform3]]`	Defines a transform set. There are complex rules defining which entries you can use for the transform arguments. These rules are explained in the command description for the **crypto ipsec transform-set** command, and Table 25 provides a list of allowed transform combinations. This command puts you into the crypto transform configuration mode.
Step 2	`Router(cfg-crypto-tran)# mode [tunnel | transport]`	(Optional) Changes the mode associated with the transform set. The mode setting is only applicable to traffic whose source and destination addresses are the IPSec peer addresses; it is ignored for all other traffic. (All other traffic is in tunnel mode only.)
Step 3	`Router(cfg-crypto-tran)# exit`	Exits the crypto transform configuration mode.

Step 4	`Router(config)# clear crypto sa`
	or
	`Router(config)# clear crypto sa peer {ip-address | peer-name}`
	or
	`Router(config)# clear crypto sa map map-name`
	or
	`Router(config)# clear crypto sa entry destination-address protocol spi`
		Clears existing IPSec security associations so that any changes to a transform set will take effect on subsequently established security associations. (Manually established SAs are reestablished immediately.)
		Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, see the **clear crypto sa** command.

Table 25 shows allowed transform combinations.

Table 25 Allowed Transform Combinations

Transform Type	Transform	Description
AH Transform *(Pick up to one.)*		
	ah-md5-hmac	AH with the MD5 (HMAC variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (HMAC variant) authentication algorithm
	ah-sha-hmac	AH with the SHA (HMAC variant) authentication algorithm

Table 25 Allowed Transform Combinations (continued)

Transform Type	Transform	Description
ESP Encryption Transform *(Pick up to one.)*		
	esp-des	ESP with the 56-bit DES encryption algorithm
	esp-3des	ESP with the 168-bit DES encryption algorithm (3DES or Triple DES)
	esp-null	Null encryption algorithm
ESP Authentication Transform *(Pick up to one.)*		
	esp-md5-hmac	ESP with the MD5 (HMAC variant) authentication algorithm
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
IP Compression Transform *(Pick up to one.)*		
	comp-lzs	IP compression with the LZS algorithm.

Creating Crypto Map Entries

To create crypto map entries, follow the guidelines and tasks described in these sections:

- About Crypto Maps
- Load Sharing
- How Many Crypto Maps Should You Create?
- Creating Crypto Map Entries to Establish Manual Security Associations
- Creating Crypto Map Entries that Use IKE to Establish Security Associations
- Creating Dynamic Crypto Maps

About Crypto Maps

Crypto map entries created for IPSec pull together the various parts used to set up IPSec security associations, including:

- Which traffic should be protected by IPSec (per a crypto access list)
- The granularity of the flow to be protected by a set of security associations
- Where IPSec-protected traffic should be sent (who the remote IPSec peer is)
- The local address to be used for the IPSec traffic (See the "Applying Crypto Map Sets to Interfaces" section for more details.)
- What IPSec security should be applied to this traffic (selecting from a list of one or more transform sets)

- Whether security associations are manually established or are established via IKE
- Other parameters that might be necessary to define an IPSec security association

Crypto map entries with the same crypto map name (but different map sequence numbers) are grouped into a crypto map set. Later, you will apply these crypto map sets to interfaces; then, all IP traffic passing through the interface is evaluated against the applied crypto map set. If a crypto map entry sees outbound IP traffic that should be protected and the crypto map specifies the use of IKE, a security association is negotiated with the remote peer according to the parameters included in the crypto map entry; otherwise, if the crypto map entry specifies the use of manual security associations, a security association should have already been established via configuration. (If a dynamic crypto map entry sees outbound traffic that should be protected and no security association exists, the packet is dropped.)

The policy described in the crypto map entries is used during the negotiation of security associations. If the local router initiates the negotiation, it will use the policy specified in the static crypto map entries to create the offer to be sent to the specified IPSec peer. If the IPSec peer initiates the negotiation, the local router will check the policy from the static crypto map entries, as well as any referenced dynamic crypto map entries to decide whether to accept or reject the peer's request (offer).

For IPSec to succeed between two IPSec peers, both peers' crypto map entries must contain compatible configuration statements.

When two peers try to establish a security association, they must each have at least one crypto map entry that is compatible with one of the other peer's crypto map entries. For two crypto map entries to be compatible, they must at least meet the following criteria:

- The crypto map entries must contain compatible crypto access lists (for example, mirror image access lists). In the case where the responding peer is using dynamic crypto maps, the entries in the local crypto access list must be "permitted" by the peer's crypto access list.
- The crypto map entries must each identify the other peer (unless the responding peer is using dynamic crypto maps).
- The crypto map entries must have at least one transform set in common.

Load Sharing

You can define multiple remote peers using crypto maps to allow for load sharing. If one peer fails, there will still be a protected path. The peer that packets are actually sent to is determined by the last peer that the router heard from (received either traffic or a negotiation request from) for a given data flow. If the attempt fails with the first peer, IKE tries the next peer on the crypto map list.

If you are not sure how to configure each crypto map parameter to guarantee compatibility with other peers, you might consider configuring dynamic crypto maps as described in the "Creating Dynamic Crypto Maps" section. Dynamic crypto maps are useful when the establishment of the IPSec tunnels is initiated by the remote peer (such as in the case of an IPSec router fronting a server). They are not useful if the establishment of the IPSec tunnels is locally initiated, because the dynamic crypto maps are policy templates, not complete statements of policy. (Although the access lists in any referenced dynamic crypto map entry are used for crypto packet filtering.)

How Many Crypto Maps Should You Create?

You can apply only one crypto map set to a single interface. The crypto map set can include a combination of IPSec/IKE and IPSec/manual entries. Multiple interfaces can share the same crypto map set if you want to apply the same policy to multiple interfaces.

If you create more than one crypto map entry for a given interface, use the *seq-num* of each map entry to rank the map entries: the lower the *seq-num*, the higher the priority. At the interface that has the crypto map set, traffic is evaluated against higher priority map entries first.

You must create multiple crypto map entries for a given interface if any of the following conditions exist:

- If different data flows are to be handled by separate IPSec peers.
- If you want to apply different IPSec security to different types of traffic (to the same or separate IPSec 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 the different types of traffic should have been defined in two separate access lists, and you must create a separate crypto map entry for each crypto access list.
- If you are not using IKE to establish a particular set of security associations, and want to specify multiple access list entries, you must create separate access lists (one per **permit** entry) and specify a separate crypto map entry for each access list.

Creating Crypto Map Entries to Establish Manual Security Associations

The use of manual security associations is a result of a prior arrangement between the users of the local router and the IPSec peer. The two parties may wish to begin with manual security associations, and then move to using security associations established via IKE, or the remote party's system may not support IKE. If IKE is not used for establishing the security associations, there is no negotiation of security associations, so the configuration information in both systems must be the same in order for traffic to be processed successfully by IPSec.

The local router can simultaneously support manual and IKE-established security associations, even within a single crypto map set. There is very little reason to disable IKE on the local router (unless the router only supports manual security associations, which is unlikely).

To create crypto map entries to establish manual security associations (SAs) (that is, when IKE is not used to establish the SAs), use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num ipsec-manual`	Specifies the crypto map entry to create (or modify). This command puts you into the crypto map configuration mode.
Step 2	`Router(config-crypto-m)# match address access-list-id`	Names an IPSec access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry. (The access list can specify only one **permit** entry when IKE is not used.)
Step 3	`Router(config-crypto-m)# set peer {hostname | ip-address}`	Specifies the remote IPSec peer. This is the peer to which IPSec protected traffic should be forwarded. (Only one peer can be specified when IKE is not used.)

	Command	Purpose
Step 4	`Router(config-crypto-m)# set transform-set transform-set-name`	Specifies which transform set should be used. This must be the same transform set that is specified in the remote peer's corresponding crypto map entry. (Only one transform set can be specified when IKE is not used.)

Step 5	`Router(config-crypto-m)# set session-key inbound ah spi hex-key-string`
	and
	`Router(config-crypto-m)# set session-key outbound ah spi hex-key-string`
		Sets the AH Security Parameter Indexes (SPIs) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the AH protocol.
		(This manually specifies the AH security association to be used with protected traffic.)

Step 6	`Router(config-crypto-m)# set session-key inbound esp spi cipher hex-key-string [authenticator hex-key-string]`
	and
	`Router(config-crypto-m)# set session-key outbound esp spi cipher hex-key-string [authenticator hex-key-string]`
		Sets the ESP Security Parameter Indexes (SPIs) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the ESP protocol. Specifies the cipher keys if the transform set includes an ESP cipher algorithm. Specifies the authenticator keys if the transform set includes an ESP authenticator algorithm.
		(This manually specifies the ESP security association to be used with protected traffic.)

Step 7	`Router(config-crypto-m)# exit`	Exits crypto-map configuration mode and return to global configuration mode.

Repeat these steps to create additional crypto map entries as required.

Creating Crypto Map Entries that Use IKE to Establish Security Associations

When IKE is used to establish security associations, the IPSec peers can negotiate the settings they will use for the new security associations. This means that you can specify lists (such as lists of acceptable transforms) within the crypto map entry.

To create crypto map entries that will use IKE to establish the security associations, use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto map map-name seq-num ipsec-isakmp`	Names the crypto map entry to create (or modify). This command puts you into the crypto map configuration mode.
Step 2	`Router(config-crypto-m)# match address access-list-id`	Names an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry.
Step 3	`Router(config-crypto-m)# set peer {hostname | ip-address}`	Specifies a remote IPSec peer. This is the peer to which IPSec protected traffic can be forwarded. Repeat for multiple remote peers.

	Command	Purpose
Step 4	`Router(config-crypto-m)# set transform-set transform-set-name1 [transform-set-name2...transform-set-name6]`	Specifies which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first).

Step 5	`Router(config-crypto-m)# set security-association lifetime seconds seconds`
	and
	`Router (config-crypto-m)# set security-association lifetime kilobytes kilobytes`
		(Optional) Specifies a security association lifetime for the crypto map entry.
		Use this command if you want the security associations for this crypto map entry to be negotiated using different IPSec security association lifetimes than the global lifetimes.

Step 6	`Router(config-crypto-m)# set security-association level per-host`	(Optional) Specifies that separate security associations should be established for each source/destination host pair. Without this command, a single IPSec "tunnel" could carry traffic for multiple source hosts and multiple destination hosts. With this command, when the router requests new security associations it will establish one set for traffic between Host A and Host B, and a separate set for traffic between Host A and Host C. Use this command with care, as multiple streams between given subnets can rapidly consume resources.
Step 7	`Router(config-crypto-m)# set pfs [group1 | group2]`	(Optional) Specifies that IPSec should ask for perfect forward secrecy when requesting new security associations for this crypto map entry, or should demand PFS in requests received from the IPSec peer.
Step 8	`Router(config-crypto-m)# exit`	Exits crypto-map configuration mode and return to global configuration mode.

Repeat these steps to create additional crypto map entries as required.

Creating Dynamic Crypto Maps

Dynamic crypto maps (this requires IKE) can ease IPSec configuration and are recommended for use with networks where the peers are not always predetermined. An example of this is mobile users, who obtain dynamically-assigned IP addresses. First, the mobile clients need to authenticate themselves to the local router's IKE by something other than an IP address, such as a fully qualified domain name. Once authenticated, the security association request can be processed against a dynamic crypto map which is set up to accept requests (matching the specified local policy) from previously unknown peers.

To configure dynamic crypto maps, follow the instructions in these sections:

- Understanding Dynamic Crypto Maps
- Creating a Dynamic Crypto Map Set
- Adding the Dynamic Crypto Map Set into a Regular (Static) Crypto Map Set

Understanding Dynamic Crypto Maps

Dynamic crypto maps are only available for use by IKE.

A dynamic crypto map entry is essentially a crypto map entry 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 a remote peer's requirements. This allows remote peers to exchange IPSec traffic with the router even if the router does not have a crypto map entry specifically configured to meet all of the remote peer's requirements.

Dynamic crypto maps are not used by the router to initiate new IPSec security associations with remote peers. Dynamic crypto maps are used when a remote peer tries to initiate an IPSec security association with the router. Dynamic crypto maps are also used in evaluating traffic.

A dynamic crypto map set is included by reference as part of a crypto map set. Any crypto map entries that reference dynamic crypto map sets should be the lowest priority crypto map entries in the crypto map set (that is, have the highest sequence numbers) so that the other crypto map entries are evaluated first; that way, the dynamic crypto map set is examined only when the other (static) map entries are not successfully matched.

If the router accepts the peer's request, at the point that it installs the new IPSec security associations it also installs a temporary crypto map entry. This entry is filled in with the results of the negotiation. At this point, the router performs normal processing, using this temporary crypto map entry as a normal entry, even requesting new security associations if the current ones are expiring (based upon the policy specified in the temporary crypto map entry). Once the flow expires (that is, all of the corresponding security associations expire), the temporary crypto map entry is then removed.

For both static and dynamic crypto maps, if unprotected inbound traffic matches a **permit** statement in an access list, and the corresponding crypto map entry is tagged as "IPSec," then the traffic is dropped because it is not IPSec-protected. (This is because the security policy as specified by the crypto map entry states that this traffic must be IPSec-protected.)

For static crypto map entries, if outbound traffic matches a **permit** statement in an access list and the corresponding SA is not yet established, the router will initiate new SAs with the remote peer. In the case of dynamic crypto map entries, if no SA existed, the traffic would simply be dropped (because dynamic crypto maps are not used for initiating new SAs).

Note Use care when using the **any** keyword in **permit** entries in dynamic crypto maps. If it is possible for the traffic covered by such a **permit** entry to include multicast or broadcast traffic, the access list should include **deny** entries for the appropriate address range. Access lists should also include **deny** entries for network and subnet broadcast traffic, and for any other traffic that should not be IPSec protected.

Creating a Dynamic Crypto Map Set

Dynamic crypto map entries, like regular static crypto map entries, are grouped into sets. A set is a group of dynamic crypto map entries all with the same *dynamic-map-name* but each with a different *dynamic-seq-num*.

To create a dynamic crypto map entry, use the following commands starting in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# crypto dynamic-map dynamic-map-name dynamic-seq-num`	Creates a dynamic crypto map entry.
Step 2	`Router(config-crypto-m)# set transform-set transform-set-name1 [transform-set-name2...transform-set-name6]`	Specifies which transform sets are allowed for the crypto map entry. List multiple transform sets in order of priority (highest priority first). This is the only configuration statement required in dynamic crypto map entries.
Step 3	`Router(config-crypto-m)# match address access-list-id`	(Optional) Accesses list number or name of an extended access list. This access list determines which traffic should be protected by IPSec and which traffic should not be protected by IPSec security in the context of this crypto map entry. **Note** Although access-lists are optional for dynamic crypto maps, they are highly recommended If this is configured, the data flow identity proposed by the IPSec peer must fall within a **permit** statement for this crypto access list. If this is not configured, the router will accept any data flow identity proposed by the IPSec peer. However, if this is configured but the specified access list does not exist or is empty, the router will drop all packets. This is similar to static crypto maps because they also require that an access list be specified. Care must be taken if the **any** keyword is used in the access list, because the access list is used for packet filtering as well as for negotiation.
Step 4	`Router(config-crypto-m)# set peer {hostname | ip-address}`	(Optional) Specifies a remote IPSec peer. Repeat for multiple remote peers. This is rarely configured in dynamic crypto map entries. Dynamic crypto map entries are often used for unknown remote peers.

Step 5	`Router(config-crypto-m)# set security-association lifetime seconds seconds`
	and
	`Router (config-crypto-m)# set security-association lifetime kilobytes kilobytes`
		(Optional) If you want the security associations for this crypto map to be negotiated using shorter IPSec security association lifetimes than the globally specified lifetimes, specify a key lifetime for the crypto map entry.

	Command	Purpose
Step 6	`Router(config-crypto-m)# set pfs [group1 | group2]`	(Optional) Specifies that IPSec should ask for perfect forward secrecy when requesting new security associations for this crypto map entry or should demand perfect forward secrecy in requests received from the IPSec peer.
Step 7	`Router(config-crypto-m)# exit`	Exits crypto-map configuration mode and return to global configuration mode.

Dynamic crypto map entries specify crypto access lists that limit traffic for which IPSec security associations can be established. A dynamic crypto map entry that does not specify an access list will be ignored during traffic filtering. A dynamic crypto map entry with an empty access list causes traffic to be dropped. If there is only one dynamic crypto map entry in the crypto map set, it must specify acceptable transform sets.

Adding the Dynamic Crypto Map Set into a Regular (Static) Crypto Map Set

You can add one or more dynamic crypto map sets into a crypto map set, via crypto map entries that reference the dynamic crypto map sets. You should set the crypto map entries referencing dynamic maps to be the lowest priority entries in a crypto map set (that is, have the highest sequence numbers).

To add a dynamic crypto map set into a crypto map set, use the following command in global configuration mode:

Command	Purpose
`Router(config)# crypto map` *map-name seq-num* `ipsec-isakmp dynamic` *dynamic-map-name*	Adds a dynamic crypto map set to a static crypto map set.

Applying Crypto Map Sets to Interfaces

You need to apply a crypto map set to each interface through which IPSec traffic will flow. Applying the crypto map set to an interface instructs the router to evaluate all the interface's traffic against the crypto map set and to use the specified policy during connection or security association negotiation on behalf of traffic to be protected by crypto.

To apply a crypto map set to an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# crypto map` *map-name*	Applies a crypto map set to an interface.

For redundancy, you could apply the same crypto map set to more than one interface. The default behavior is as follows:

- Each interface will have its own piece of the security association database.
- The IP address of the local interface will be used as the local address for IPSec traffic originating from or destined to that interface.

If you apply the same crypto map set to multiple interfaces for redundancy purposes, you need to specify an identifying interface. This has the following effects:

- The per-interface portion of the IPSec security association database will be established one time and shared for traffic through all the interfaces that share the same crypto map.
- The IP address of the identifying interface will be used as the local address for IPSec traffic originating from or destined to those interfaces sharing the same crypto map set.

One suggestion is to use a loopback interface as the identifying interface.

To specify redundant interfaces and name an identifying interface, use the following command in global configuration mode:

Command	Purpose
`Router(config)# crypto map map-name local-address interface-id`	Permits redundant interfaces to share the same crypto map, using the same local identity.

Monitoring and Maintaining IPSec

Certain configuration changes will only take effect when negotiating subsequent security associations. If you want the new settings to take immediate effect, you must clear the existing security associations so that they will be re-established with the changed configuration. For manually established security associations, you must clear and reinitialize the security associations or the changes will never take effect. If the router is actively processing IPSec traffic, it is desirable to clear only the portion of the security association database that would be affected by the configuration changes (that is, clear only the security associations established by a given crypto map set). Clearing the full security association database should be reserved for large-scale changes, or when the router is processing very little other IPSec traffic.

To clear (and reinitialize) IPSec security associations, use one of the following commands in global configuration mode:

Command	Purpose

`Router(config)# clear crypto sa`
or
`Router(config)# clear crypto sa peer {ip-address | peer-name}`
or
`Router(config)# clear crypto sa map map-name`
or
`Router(config)# clear crypto sa entry destination-address protocol spi`
	Clears IPSec security associations.
	Note Using the **clear crypto sa** command without parameters will clear out the full SA database, which will clear out active security sessions. You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database. For more information, see the **clear crypto sa** command.

To view information about your IPSec configuration, use one or more of the following commands in EXEC mode:

Command	Purpose
Router# **show crypto ipsec transform-set**	Displays your transform set configuration.
Router# **show crypto map** [**interface** *interface* | **tag** *map-name*]	Displays your crypto map configuration.
Router# **show crypto ipsec sa** [**map** *map-name* | **address** | **identity**] [**detail**]	Displays information about IPSec security associations.
Router# **show crypto dynamic-map** [**tag** *map-name*]	Displays information about dynamic crypto maps.
Router# **show crypto ipsec security-association lifetime**	Displays global security association lifetime values.

IPSec Configuration Example

The following example shows a minimal IPSec configuration where the security associations will be established via IKE. For more information about IKE, see the "Configuring Internet Key Exchange Security Protocol" chapter.

An IPSec access list defines which traffic to protect:

```
access-list 101 permit ip 10.0.0.0 0.0.0.255 10.2.2.0 0.0.0.255
```

A transform set defines how the traffic will be protected. In this example, transform set "myset1" uses DES encryption and SHA for data packet authentication:

```
crypto ipsec transform-set myset1 esp-des esp-sha
```

Another transform set example is "myset2," which uses Triple DES encryptions and MD5 (HMAC variant) for data packet authentication:

```
crypto ipsec transform-set myset2 esp-3des esp-md5-hmac
```

A crypto map joins together the IPSec access list and transform set and specifies where the protected traffic is sent (the remote IPSec peer):

```
crypto map toRemoteSite 10 ipsec-isakmp
 match address 101
 set transform-set myset2
 set peer 10.2.2.5
```

The crypto map is applied to an interface:

```
interface Serial0
 ip address 10.0.0.2
 crypto map toRemoteSite
```

Note In this example, IKE must be enabled.