



Configuring Security for VPNs with IPsec

This module describes how to configure basic IPsec VPNs. IPsec is a framework of open standards developed by the IETF. It provides security for the 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.



Note

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

- [Finding Feature Information](#), on page 1
- [Prerequisites for Configuring Security for VPNs with IPsec](#), on page 2
- [Restrictions for Configuring Security for VPNs with IPsec](#), on page 2
- [Information About Configuring Security for VPNs with IPsec](#), on page 3
- [How to Configure IPsec VPNs](#), on page 10
- [Configuration Examples for IPsec VPN](#), on page 26
- [Additional References for Configuring Security for VPNs with IPsec](#), on page 27
- [Feature Information for Configuring Security for VPNs with IPsec](#), on page 29
- [Glossary](#), on page 30

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for Configuring Security for VPNs with IPsec

IKE Configuration

You must configure Internet Key Exchange (IKE) as described in the module *Configuring Internet Key Exchange for IPsec VPNs*.



Note

If you decide not to use IKE, you must still disable it as described in the module *Configuring Internet Key Exchange for IPsec VPNs*.

Ensure Access Lists Are Compatible with IPsec

IKE uses UDP port 500. The IPsec encapsulating security payload (ESP) and authentication header (AH) protocols use protocol numbers 50 and 51, respectively. Ensure that your access lists are configured so that traffic from protocol 50, 51, and UDP port 500 are 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.

Restrictions for Configuring Security for VPNs with IPsec

Cisco IPsec Policy Map MIB

The MIB OID objects are displayed only when an IPsec session is up.

Discontiguous Access Control Lists

Crypto maps using access control lists (ACLs) that have discontiguous masks are not supported.

IPv4 Packets with IP Options Set

The following platforms do not support encrypting IPv4 packets with IP options set:

Cisco ASR1001 and ASR1000 routers with ESP-5, ESP-10, ESP-20, and ESP-40.

Physical Interface and Crypto Map

A crypto map on a physical interface is not supported, if the physical interface is the source interface of a tunnel protection interface.

NAT Configuration

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

Unicast IP Datagram Application Only

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.

IPv6 BFD with ISM

IPv6 BFD with ISM is not supported on ISRG2 routers.

Unsupported Interface Types

- Crypto VPNs are not supported on the bridge domain interfaces (BDI).
- Crypto maps are not supported on tunnel interface and port-channel interface.

Information About Configuring Security for VPNs with IPsec

Supported Standards

Cisco implements the following standards with this feature:

- IPsec—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; IPsec uses IKE to handle negotiation of protocols and algorithms based on the local policy, and 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.

- IKE (IKEv1 and IKEv2)—A hybrid protocol that implements Oakley and SKEME key exchanges inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. While IKE is used with other protocols, its initial implementation is with the IPsec protocol. IKE provides authentication of IPsec peers, negotiates IPsec security associations, and establishes IPsec keys.

The component technologies implemented for IPsec include:



Note Cisco no longer recommends using DES, 3DES, MD5 (including HMAC variant), and Diffie-Hellman (DH) groups 1, 2 and 5; instead, you should use AES, SHA and DH Groups 14 or higher. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption \(NGE\)](#) white paper.

- AES—Advanced Encryption Standard. A cryptographic algorithm that protects sensitive, unclassified information. AES is a privacy transform for IPsec and IKE and has been developed to replace DES. AES

is designed to be more secure than DES. AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length—the algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.

- **DES**—Data Encryption Standard. An algorithm that is used to encrypt packet data. Cisco software 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. Cisco no longer recommends Triple DES (3DES).

**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 an e-mail to export@cisco.com.

- **SEAL**—Software Encryption Algorithm. An alternative algorithm to software-based DES, 3DES, and AES. SEAL encryption uses a 160-bit encryption key and has a lower impact on the CPU when compared to other software-based algorithms.
- **SHA-2 and SHA-1 family (HMAC variant)**—Secure Hash Algorithm (SHA) 1 and 2. Both SHA-1 and SHA-2 are hash algorithms used to authenticate packet data and verify the integrity verification mechanisms for the IKE protocol. HMAC is a variant that provides an additional level of hashing. SHA-2 family adds the SHA-256 bit hash algorithm and SHA-384 bit hash algorithm. This functionality is part of the Suite-B requirements that comprises four user interface suites of cryptographic algorithms for use with IKE and IPsec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. See the Configuring Security for VPNs with IPsec feature module for more detailed information about Cisco IOS Suite-B support.
- **Diffie-Hellman**—A public-key cryptography protocol that allows two parties to establish a shared secret over an unsecure communications channel. Diffie-Hellman is used within IKE to establish session keys. It supports 768-bit (the default), 1024-bit, 1536-bit, 2048-bit, 3072-bit, and 4096-bit DH groups. It also supports a 2048-bit DH group with a 256-bit subgroup, and 256-bit and 384-bit elliptic curve DH (ECDH). Cisco recommends using 2048-bit or larger DH key exchange, or ECDH key exchange.
- **MD5 (Hash-based Message Authentication Code (HMAC) variant)**—Message digest algorithm 5 (MD5) is a hash algorithm. HMAC is a keyed hash variant used to authenticate data.

IPsec as implemented in Cisco 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.

Supported Encapsulation

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

IPsec also works with Generic Routing Encapsulation (GRE) and IPinIP Layer 3, Layer 2 Forwarding (L2F), Layer 2 Tunneling Protocol (L2TP), Data Link Switching+ (DLSw+), and Source Route Bridging (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).

IPsec Functionality Overview

IPsec provides the following network security services. (In general, the local security policy dictates 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 sent IPsec packets. This service is dependent upon the data integrity service.
- Anti-replay—The IPsec receiver can detect and reject replayed packets.

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 that should be used to protect these sensitive packets by specifying the characteristics of these tunnels. When the IPsec peer recognizes a sensitive packet, the peer sets up the appropriate secure tunnel and sends the packet through the tunnel to the remote peer. (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 (SAs) that are established between two IPsec peers. The SAs define the protocols and algorithms to be applied to sensitive packets and specify the keying material to be used by the two peers. SAs are unidirectional and are established per security protocol (AH or ESP).

Multiple IPsec tunnels can exist between two peers to secure different data streams, with each tunnel using a separate set of SAs. For example, some data streams only need to be authenticated, while other data streams must both be encrypted and authenticated.

IKEv1 Transform Sets

An Internet Key Exchange version 1 (IKEv1) transform set represents a certain combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

IKEv2 Transform Sets

An Internet Key Exchange version 2 (IKEv2) proposal is a set of transforms used in the negotiation of IKEv2 SA as part of the IKE_SA_INIT exchange. An IKEv2 proposal is regarded as complete only when it has at

least an encryption algorithm, an integrity algorithm, and a Diffie-Hellman (DH) group configured. If no proposal is configured and attached to an IKEv2 policy, then the default proposal is used in the negotiation. The default proposal is a collection of commonly used algorithms which are as follows:

```
encryption aes-cbc-128 3des
integrity sha1 md5
group 5 2
```

Although the **crypto ikev2 proposal** command is similar to the **crypto isakmp policy priority** command, the IKEv2 proposal differs as follows:

- An IKEv2 proposal allows configuration of one or more transforms for each transform type.
- An IKEv2 proposal does not have any associated priority.


Note

To use IKEv2 proposals in negotiation, they must be attached to IKEv2 policies. If a proposal is not configured, then the default IKEv2 proposal is used with the default IKEv2 policy.

Transform Sets: A Combination of Security Protocols and Algorithms

About Transform Sets


Note

Cisco no longer recommends using ah-md5-hmac, esp-md5-hmac, esp-des or esp-3des. Instead, you should use ah-sha-hmac, esp-sha-hmac or esp-aes. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

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

During IPsec security association negotiations with IKE, peers search for an identical transform set for both peers. When such a transform set is found, it is selected and applied to the protected traffic as part of both peers' IPsec SAs. (With manually established SAs, there is no negotiation with the peer, so both sides must specify the same transform set.)

The table below shows allowed transform combinations.

Table 1: Allowed Transform Combinations

Transform Type	Transform	Description
AH Transform (Pick only one.)	ah-md5-hmac	AH with the MD5 (Message Digest 5) (an HMAC variant) authentication algorithm. (No longer recommended).
	ah-sha-hmac	AH with the SHA (Secure Hash Algorithm) (an HMAC variant) authentication algorithm.

Transform Type	Transform	Description
ESP Encryption Transform (<i>Pick only one.</i>)	esp-aes	ESP with the 128-bit Advanced Encryption Standard (AES) encryption algorithm.
	esp-gcm esp-gmac	<p>The esp-gcm and esp-gmac transforms are ESPs with either a 128-bit or a 256-bit encryption algorithm. The default for either of these transforms is 128 bits.</p> <p>Both esp-gcm and esp-gmac transforms cannot be configured together with any other ESP transform within the same crypto IPsec transform set using the crypto ipsec transform-set command.</p> <p>The esp-gcm and esp-gmac combinations are not supported on the Cisco ASR 1001 routers with the following ESPs:</p> <ul style="list-style-type: none"> • ESP-5 • ESP-10 • ESP-20 • ESP-40
	esp-aes 192	ESP with the 192-bit AES encryption algorithm.
	esp-aes 256	ESP with the 256-bit AES encryption algorithm.
	esp-des	ESP with the 56-bit Data Encryption Standard (DES) encryption algorithm. (No longer recommended).
	esp-3des	ESP with the 168-bit DES encryption algorithm (3DES or Triple DES). (No longer recommended).
	esp-null	Null encryption algorithm.
	esp-seal	ESP with the 160-bit SEAL encryption algorithm.
ESP Authentication Transform (<i>Pick only one.</i>)	esp-md5-hmac	ESP with the MD5 (HMAC variant) authentication algorithm. (No longer recommended).
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm.
IP Compression Transform	comp-lzs	IP compression with the Lempel-Ziv-Stac (LZS) algorithm



Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

Cisco IOS Suite-B Support for IKE and IPsec Cryptographic Algorithms

Suite-B adds support for four user interface suites of cryptographic algorithms for use with IKE and IPsec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm.

Suite-B is not supported on the following hardware platforms:

- Cisco ASR1001
- ESP-5
- ESP-10
- ESP-20
- ESP-40

Suite-B has the following cryptographic algorithms:

- Suite-B-GCM-128-Provides ESP integrity protection, confidentiality, and IPsec encryption algorithms that use the 128-bit AES using Galois and Counter Mode (AES-GCM) described in RFC 4106. This suite should be used when ESP integrity protection and encryption are both needed.
- Suite-B-GCM-256-Provides ESP integrity protection and confidentiality using 256-bit AES-GCM described in RFC 4106. This suite should be used when ESP integrity protection and encryption are both needed.
- Suite-B-GMAC-128-Provides ESP integrity protection using 128-bit AES- Galois Message Authentication Code (GMAC) described in RFC 4543, but does not provide confidentiality. This suite should be used only when there is no need for ESP encryption.
- Suite-B-GMAC-256-Provides ESP integrity protection using 256-bit AES-GMAC described in RFC 4543, but does not provide confidentiality. This suite should be used only when there is no need for ESP encryption.

IPsec encryption algorithms use AES-GCM when encryption is required and AES-GMAC for message integrity without encryption.

IKE negotiation uses AES Cipher Block Chaining (CBC) mode to provide encryption and Secure Hash Algorithm (SHA)-2 family containing the SHA-256 and SHA-384 hash algorithms, as defined in RFC 4634, to provide the hash functionality. Diffie-Hellman using Elliptic Curves (ECP), as defined in RFC 4753, is used for key exchange and the Elliptic Curve Digital Signature Algorithm (ECDSA), as defined in RFC 4754, to provide authentication.

Suite-B Requirements

Suite-B imposes the following software crypto engine requirements for IKE and IPsec:

- HMAC-SHA256 and HMAC-SHA384 are used as pseudorandom functions; the integrity check within the IKE protocol is used. Optionally, HMAC-SHA512 can be used.
- Elliptic curve groups 19 (256-bit ECP curve) and 20 (384-bit ECP curve) are used as the Diffie-Hellman group in IKE. Optionally, group 21 (521-bit ECP curve) can be used.
- The Elliptic Curve Digital Signature Algorithm (ECDSA) algorithm (256-bit and 384-bit curves) is used for the signature operation within X.509 certificates.

- GCM (16 byte ICV) and GMAC is used for ESP (128-bit and 256-bit keys). Optionally, 192-bit keys can be used.
- Public Key Infrastructure (PKI) support for validation of X.509 certificates using ECDSA signatures must be used.
- PKI support for generating certificate requests using ECDSA signatures and for importing the issued certificates into IOS must be used.
- IKEV2 support for allowing the ECDSA signature (ECDSA-sig) as authentication method must be used.

Where to Find Suite-B Configuration Information

Suite-B configuration support is described in the following documents:

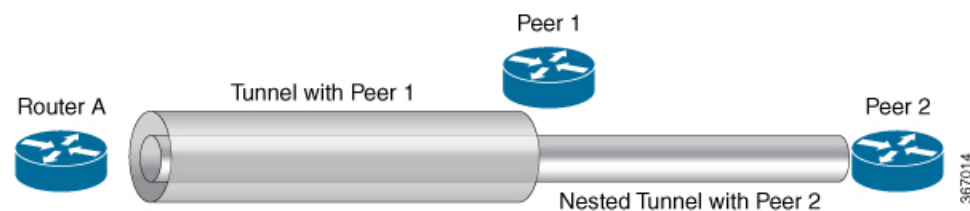
- For more information on SHA-2 family (HMAC variant) and Elliptic Curve (EC) key pair configuration, see the *Configuring Internet Key Exchange for IPsec VPNs* feature module.
- For more information on configuring a transform for an integrity algorithm type, see the “Configuring the IKEv2 Proposal” section in the *Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site* feature module.
- For more information on configuring the ECDSA-sig to be the authentication method for IKEv2, see the “Configuring IKEv2 Profile (Basic)” section in the *Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site* feature module.
- For more information on configuring elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation, see the *Configuring Internet Key Exchange for IPsec VPNs* and *Configuring Internet Key Exchange Version 2 and FlexVPN* feature modules.

For more information on the Suite-B support for certificate enrollment for a PKI, see the *Configuring Certificate Enrollment for a PKI* feature module.

Nested IPsec Tunnels

The following figure illustrates nested IPsec tunnels, where a tunnel is transported inside another tunnel.

Figure 1: Nested IPsec Tunnels



IPsec supports nested tunnels, where a tunnel is transported inside another tunnel. For example, in the above topology Router A has IPsec tunnels with Peer 1 and Peer 2, wherein the tunnel with Peer 2 is transported inside the tunnel with Peer 1. The following are sample configurations to establish nested IPsec tunnels on Router A, Peer 1 and Peer 2:

Configuration snippet on Router A

```
interface Tunnel1
ip address 172.16.0.1 255.255.255.0
```

```
tunnel source FastEthernet0/0
tunnel destination 10.0.0.2
tunnel protection ipsec profile prof1
!
interface Tunnel2
ip address 192.168.1.1 255.255.255.0
tunnel source Tunnel1
tunnel destination 10.0.1.2
tunnel mode ipsec ipv4
tunnel protection ipsec profile prof2
```

Configuration snippet on Peer 1

```
interface Tunnel1
ip address 172.16.0.2 255.255.255.0
tunnel source FastEthernet0/0.110
tunnel destination 10.0.0.1
tunnel protection ipsec profile prof1
```

Configuration snippet on Peer 2

```
interface Tunnel2
ip address 192.168.1.2 255.255.255.0
tunnel source GigabitEthernet0/1
tunnel destination 172.16.0.1
tunnel mode ipsec ipv4
tunnel protection ipsec profile prof2
```

The following nested tunnel combinations are supported with **tunnel protection** command:

- Virtual tunnel interface (VTI) in VTI
- VTI in generic routing encapsulation (GRE) or IPsec
- GRE or IPsec in GRE or IPsec
- GRE or IPsec in VTI

How to Configure IPsec VPNs

Creating Crypto Access Lists

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
 - **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**]
 - **ip access-list extended** *name*
4. Repeat Step 3 for each crypto access list you want to create.

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	Do one of the following: <ul style="list-style-type: none"> access-list <i>access-list-number</i> {deny permit} <i>protocol source source-wildcard destination destination-wildcard</i> [log] ip access-list extended <i>name</i> Example: Device(config)# access-list 100 permit ip 10.0.68.0 0.0.0.255 10.1.1.0 0.0.0.255 Example: Device(config)# ip access-list extended vpn-tunnel	Specifies conditions to determine which IP packets are protected. <ul style="list-style-type: none"> 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. Enable or disable crypto for traffic that matches these conditions. Tip Cisco recommends that you configure “mirror image” crypto access lists for use by IPsec and that you avoid using the any keyword.
Step 4	Repeat Step 3 for each crypto access list you want to create.	—

What to Do Next

After at least one crypto access list is created, a transform set needs to be defined as described in the [“Configuring Transform Sets for IKEv1 and IKEv2 Proposals”](#) section.

Next the crypto access lists need to be associated to particular interfaces when you configure and apply crypto map sets to the interfaces. (Follow the instructions in the [“Creating Crypto Map Sets”](#) and [“Applying Crypto Map Sets to Interfaces”](#) sections).

Configuring Transform Sets for IKEv1 and IKEv2 Proposals

Perform this task to define a transform set that is to be used by the IPsec peers during IPsec security association negotiations with IKEv1 and IKEv2 proposals.

Restrictions

If you are specifying SEAL encryption, note the following restrictions:

- Your router and the other peer must not have a hardware IPsec encryption.
- Your router and the other peer must support IPsec.
- Your router and the other peer must support the k9 subsystem.

- SEAL encryption is available only on Cisco equipment. Therefore, interoperability is not possible.
- Unlike IKEv1, the authentication method and SA lifetime are not negotiable in IKEv2, and because of this, these parameters cannot be configured under the IKEv2 proposal.

Configuring Transform Sets for IKEv1

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]
4. **mode** [*tunnel* | *transport*]
5. **end**
6. **clear crypto sa** [*peer {ip-address | peer-name}* | **sa map** *map-name* | **sa entry** *destination-address protocol spi*]
7. **show crypto ipsec transform-set** [*tag transform-set-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto ipsec transform-set <i>transform-set-name transform1</i> [<i>transform2</i> [<i>transform3</i>]] Example: Device(config)# crypto ipsec transform-set aasset esp-aes 256 esp-sha-hmac	Defines a transform set and enters crypto transform configuration mode. • There are complex rules defining the entries that you can use for transform arguments. These rules are explained in the command description for the crypto ipsec transform-set command, and the table in “ About Transform Sets ” section provides a list of allowed transform combinations.
Step 4	mode [<i>tunnel</i> <i>transport</i>] Example: Device(cfg-crypto-tran)# mode transport	(Optional) Changes the mode associated with the transform set. • The mode setting is applicable only 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 5	end Example:	Exits crypto transform configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
	Device(cfg-crypto-tran)# end	
Step 6	clear crypto sa [<i>peer {ip-address peer-name} sa map map-name sa entry destination-address protocol spi</i>] Example: Device# clear crypto sa	(Optional) Clears existing IPsec security associations so that any changes to a transform set takes effect on subsequently established security associations. Manually established SAs are reestablished immediately. <ul style="list-style-type: none"> Using the clear crypto sa command without parameters clears out the full SA database, which clears out active security sessions. You may also specify the peer, map, or entry keywords to clear out only a subset of the SA database.
Step 7	show crypto ipsec transform-set [<i>tag transform-set-name</i>] Example: Device# show crypto ipsec transform-set	(Optional) Displays the configured transform sets.

What to Do Next

After you have defined a transform set, you should create a crypto map as specified in the *Creating Crypto Map Sets* section.

Configuring Transform Sets for IKEv2

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto ikev2 proposal *proposal-name*
4. encryption *transform1* [*transform2*] ...
5. integrity *transform1* [*transform2*] ...
6. group *transform1* [*transform2*] ...
7. end
8. show crypto ikev2 proposal

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	crypto ikev2 proposal <i>proposal-name</i> Example: Device(config)# crypto ikev2 proposal proposal-1	Specifies the name of the proposal and enters crypto IKEv2 proposal configuration mode. <ul style="list-style-type: none"> The proposals are referred in IKEv2 policies through the proposal name.
Step 4	encryption <i>transform1</i> [<i>transform2</i>] ... Example: Device(config-ikev2-proposal)# encryption aes-cbc-128	(Optional) Specifies one or more transforms of the following encryption type: <ul style="list-style-type: none"> AES-CBC 128—128-bit AES-CBC AES-CBC 192—192-bit AES-CBC AES-CBC 256—256-bit AES-CBC 3DES—168-bit DES (No longer recommended. AES is the recommended encryption algorithm).
Step 5	integrity <i>transform1</i> [<i>transform2</i>] ... Example: Device(config-ikev2-proposal)# integrity sha1	(Optional) Specifies one or more transforms of the following integrity type: <ul style="list-style-type: none"> The sha256 keyword specifies SHA-2 family 256-bit (HMAC variant) as the hash algorithm. The sha384 keyword specifies SHA-2 family 384-bit (HMAC variant) as the hash algorithm. The sha512 keyword specifies SHA-2 family 512-bit (HMAC variant) as the hash algorithm the sha1 keyword specifies the SHA-1 (HMAC variant) as the hash algorithm. The md5 keyword specifies MD5 (HMAC variant) as the hash algorithm. (No longer recommended. SHA-1 is the recommended replacement.)
Step 6	group <i>transform1</i> [<i>transform2</i>] ... Example: Device(config-ikev2-proposal)# group 14	(Optional) Specifies one or more transforms of the possible DH group type: <ul style="list-style-type: none"> 1—768-bit DH (No longer recommended.) 2—1024-bit DH (No longer recommended) 5—1536-bit DH (No longer recommended) 14—Specifies the 2048-bit DH group. 15—Specifies the 3072-bit DH group. 16—Specifies the 4096-bit DH group. 19—Specifies the 256-bit elliptic curve DH (ECDH) group. 20—Specifies the 384-bit ECDH group.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • 24—Specifies the 2048-bit DH/DSA group.
Step 7	end Example: Device(config-ikev2-proposal)# end	Exits crypto IKEv2 proposal configuration mode and returns to privileged EXEC mode.
Step 8	show crypto ikev2 proposal Example: Device# show crypto ikev2 proposal	(Optional) Displays the parameters for each IKEv2 proposal.

Transform Sets for IKEv2 Examples

The following examples show how to configure a proposal:

IKEv2 Proposal with One Transform for Each Transform Type

```
Device(config)# crypto ikev2 proposal proposal-1
Device(config-ikev2-proposal)# encryption aes-cbc-128
Device(config-ikev2-proposal)# integrity sha1
Device(config-ikev2-proposal)# group 14
```

IKEv2 Proposal with Multiple Transforms for Each Transform Type

```
crypto ikev2 proposal proposal-2
encryption aes-cbc-128 aes-cbc-192
integrity sha1 sha256
group 14 15
```

For a list of transform combinations, see [Configuring Security for VPNs with IPsec](#).

IKEv2 Proposals on the Initiator and Responder

The proposal of the initiator is as follows:

```
Device(config)# crypto ikev2 proposal proposal-1
Device(config-ikev2-proposal)# encryption aes-cbc-128 aes-cbc-196
Device(config-ikev2-proposal)# integrity sha1 sha256
Device(config-ikev2-proposal)# group 14 16
```

The proposal of the responder is as follows:

```
Device(config)# crypto ikev2 proposal proposal-2
Device(config-ikev2-proposal)# encryption aes-cbc-196 aes-cbc-128
Device(config-ikev2-proposal)# integrity sha256 sha1
Device(config-ikev2-proposal)# group 16 14
```

In the scenario, the initiator's choice of algorithms is preferred and the selected algorithms are as follows:

```
encryption aes-cbc-128
integrity sha1
group 14
```

What to Do Next

After you have defined a transform set, you should create a crypto map as specified in the *Creating Crypto Map Sets* section.

Creating Crypto Map Sets

Creating Static Crypto Maps

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

Perform this task to create crypto map entries that use IKE to establish SAs. To create IPv6 crypto map entries, you must use the **ipv6** keyword with the **crypto map** command. For IPv4 crypto maps, use the **crypto map** command without the **ipv6** keyword.



Note

Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map [ipv6] map-name seq-num [ipsec-isakmp]**
4. **match address access-list-id**
5. **set peer {hostname | ip-address}**
6. **crypto ipsec security-association dummy {pps rate | seconds seconds}**
7. **set transform-set transform-set-name1 [transform-set-name2...transform-set-name6]**
8. **set security-association lifetime {seconds seconds | kilobytes kilobytes | kilobytes disable}**
9. **set security-association level per-host**
10. **set pfs [group1 | group14 | group15 | group16 | group19 | group2 | group20 | group24 | group5]**
11. **end**
12. **show crypto map [interface interface | tag map-name]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
Step 3	crypto map [ipv6] map-name seq-num [ipsec-isakmp] Example: <pre>Device(config)# crypto map static-map 1 ipsec-isakmp</pre>	Creates or modifies a crypto map entry, and enters crypto map configuration mode. <ul style="list-style-type: none"> For IPv4 crypto maps, use the command without the ipv6 keyword.
Step 4	match address access-list-id Example: <pre>Device(config-crypto-m)# match address vpn-tunnel</pre>	Names an extended access list. <ul style="list-style-type: none"> This access list determines the traffic that should be protected by IPsec and the traffic that should not be protected by IPsec security in the context of this crypto map entry.
Step 5	set peer {hostname ip-address} Example: <pre>Device(config-crypto-m)# set-peer 192.168.101.1</pre>	Specifies a remote IPsec peer—the peer to which IPsec protected traffic can be forwarded. <ul style="list-style-type: none"> Repeat for multiple remote peers.
Step 6	crypto ipsec security-association dummy {pps rate seconds seconds} Example: <pre>Device(config-crypto-m)# set security-association dummy seconds 5</pre>	Enables generating dummy packets. These dummy packets are generated for all flows created in the crypto map.
Step 7	set transform-set transform-set-name1 [transform-set-name2...transform-set-name6] Example: <pre>Device(config-crypto-m)# set transform-set aasset</pre>	Specifies the transform sets that are allowed for this crypto map entry. <ul style="list-style-type: none"> List multiple transform sets in the order of priority (highest priority first).
Step 8	set security-association lifetime {seconds seconds kilobytes kilobytes kilobytes disable} Example: <pre>Device (config-crypto-m)# set security-association lifetime seconds 2700</pre>	(Optional) Specifies a SA lifetime for the crypto map entry. <ul style="list-style-type: none"> By default, the SAs of the crypto map are negotiated according to the global lifetimes, which can be disabled.
Step 9	set security-association level per-host Example: <pre>Device(config-crypto-m)# set security-association level per-host</pre>	(Optional) Specifies that separate SAs should be established for each source and destination host pair. <ul style="list-style-type: none"> By default, a single IPsec “tunnel” can carry traffic for multiple source hosts and multiple destination hosts. <p>Caution Use this command with care because multiple streams between given subnets can rapidly consume resources.</p>
Step 10	set pfs [group1 group14 group15 group16 group19 group2 group20 group24 group5]	(Optional) Specifies that IPsec either should ask for password forward secrecy (PFS) when requesting new SAs

	Command or Action	Purpose
	Example: <pre>Device(config-crypto-m)# set pfs group14</pre>	for this crypto map entry or should demand PFS in requests received from the IPsec peer. <ul style="list-style-type: none"> • Group 1 specifies the 768-bit Diffie-Hellman (DH) identifier (default). (No longer recommended). • Group 2 specifies the 1024-bit DH identifier. (No longer recommended). • Group 5 specifies the 1536-bit DH identifier. (No longer recommended) • Group 14 specifies the 2048-bit DH identifier. • Group 15 specifies the 3072-bit DH identifier. • Group 16 specifies the 4096-bit DH identifier. • Group 19 specifies the 256-bit elliptic curve DH (ECDH) identifier. • Group 20 specifies the 384-bit ECDH identifier. • Group 24 specifies the 2048-bit DH/DSA identifier • By default, PFS is not requested. If no group is specified with this command, group 1 is used as the default.
Step 11	end Example: <pre>Device(config-crypto-m)# end</pre>	Exits crypto map configuration mode and returns to privileged EXEC mode.
Step 12	show crypto map [interface <i>interface</i> tag <i>map-name</i>] Example: <pre>Device# show crypto map</pre>	Displays your crypto map configuration.

Troubleshooting Tips

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they are reestablished with the changed configuration. If the router is actively processing IPsec traffic, clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the full SA database should be reserved for large-scale changes, or when the router is processing very little other IPsec traffic.

To clear IPsec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears out the full SA database, which clears active security sessions.)

What to Do Next

After you have successfully created a static crypto map, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see the “[Applying Crypto Map Sets to Interfaces](#)” section.

Creating Dynamic Crypto Maps

Dynamic crypto map entries specify crypto access lists that limit traffic for which IPsec SAs can be established. A dynamic crypto map entry that does not specify an access list is 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 the acceptable transform sets.

Perform this task to create dynamic crypto map entries that use IKE to establish the SAs.



Note IPv6 addresses are not supported on dynamic crypto maps.



Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the [Next Generation Encryption](#) (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto dynamic-map** *dynamic-map-name* *dynamic-seq-num*
4. **set transform-set** *transform-set-name1* [*transform-set-name2*...*transform-set-name6*]
5. **match address** *access-list-id*
6. **set peer** {*hostname* | *ip-address*}
7. **set security-association lifetime** {*seconds seconds* | *kilobytes kilobytes* | *kilobytes disable*}
8. **set pfs** [*group1* | *group14* | *group15* | *group16* | *group19* | *group2* | *group20* | *group24* | *group5*]
9. **exit**
10. **exit**
11. **show crypto dynamic-map** [*tag map-name*]
12. **configure terminal**
13. **crypto map** *map-name* *seq-num* **ipsec-isakmp dynamic** *dynamic-map-name* [*discover*]
14. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto dynamic-map <i>dynamic-map-name</i> <i>dynamic-seq-num</i> Example: Device(config)# crypto dynamic-map test-map 1	Creates a dynamic crypto map entry and enters crypto map configuration mode.
Step 4	set transform-set <i>transform-set-name1</i> [<i>transform-set-name2...transform-set-name6</i>] Example: Device(config-crypto-m)# set transform-set aesset	Specifies the transform sets allowed for the crypto map entry. <ul style="list-style-type: none"> List multiple transform sets in the order of priority (highest priority first). This is the only configuration statement required in dynamic crypto map entries.
Step 5	match address <i>access-list-id</i> Example: Device(config-crypto-m)# match address 101	(Optional) Specifies the list number or name of an extended access list. <ul style="list-style-type: none"> 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. <p>Note Although access lists are optional for dynamic crypto maps, they are highly recommended.</p> <ul style="list-style-type: none"> If an access list is configured, the data flow identity proposed by the IPsec peer must fall within a permit statement for this crypto access list. If an access list is not configured, the device accepts any data flow identity proposed by the IPsec peer. However, if an access list is configured but the specified access list does not exist or is empty, the device drops all packets. This is similar to static crypto maps, which require access lists to 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. You must configure a match address; otherwise, the behavior is not secure, and you cannot enable TED because packets are sent in the clear (unencrypted.)
Step 6	set peer { <i>hostname</i> <i>ip-address</i> } Example:	(Optional) Specifies a remote IPsec peer. Repeat this step for multiple remote peers.

	Command or Action	Purpose
	Device(config-crypto-m)# set peer 192.168.101.1	Note This is rarely configured in dynamic crypto map entries. Dynamic crypto map entries are often used for unknown remote peers.
Step 7	set security-association lifetime {seconds <i>seconds</i> kilobytes <i>kilobytes</i> kilobytes disable} Example: Device(config-crypto-m)# set security-association lifetime seconds 7200	(Optional) Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IP Security SAs. Note To minimize the possibility of packet loss when rekeying in high bandwidth environments, you can disable the rekey request triggered by a volume lifetime expiry.
Step 8	set pfs [group1 group14 group15 group16 group19 group2 group20 group24 group5] Example: Device(config-crypto-m)# set pfs group14	(Optional) Specifies that IPsec should ask for PFS when requesting new security associations for this crypto map entry or should demand PFS in requests received from the IPsec peer. <ul style="list-style-type: none"> • Group 1 specifies the 768-bit Diffie-Hellman (DH) identifier (default). (No longer recommended). • Group 2 specifies the 1024-bit DH identifier. (No longer recommended). • Group 5 specifies the 1536-bit DH identifier. (No longer recommended) • Group 14 specifies the 2048-bit DH identifier. • Group 15 specifies the 3072-bit DH identifier. • Group 16 specifies the 4096-bit DH identifier. • Group 19 specifies the 256-bit elliptic curve DH (ECDH) identifier. • Group 20 specifies the 384-bit ECDH identifier. • Group 24 specifies the 2048-bit DH/DSA identifier • By default, PFS is not requested. If no group is specified with this command, group1 is used as the default.
Step 9	exit Example: Device(config-crypto-m)# exit	Exits crypto map configuration mode and returns to global configuration mode.
Step 10	exit Example: Device(config)# exit	Exits global configuration mode.

	Command or Action	Purpose
Step 11	show crypto dynamic-map [tag <i>map-name</i>] Example: Device# show crypto dynamic-map	(Optional) Displays information about dynamic crypto maps.
Step 12	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 13	crypto map <i>map-name</i> <i>seq-num</i> ipsec-isakmp dynamic <i>dynamic-map-name</i> [discover] Example: Device(config)# crypto map static-map 1 ipsec-isakmp dynamic test-map discover	(Optional) Adds a dynamic crypto map to a crypto map set. <ul style="list-style-type: none"> You should set the crypto map entries referencing dynamic maps to the lowest priority entries in a crypto map set. <p>Note You must enter the discover keyword to enable TED.</p>
Step 14	exit Example: Device(config)# exit	Exits global configuration mode.

Troubleshooting Tips

Certain configuration changes take effect only when negotiating subsequent SAs. If you want the new settings to take immediate effect, you must clear the existing SAs so that they are reestablished with the changed configuration. If the router is actively processing IPsec traffic, clear only the portion of the SA database that would be affected by the configuration changes (that is, clear only the SAs established by a given crypto map set). Clearing the entire SA database must be reserved for large-scale changes, or when the router is processing minimal IPsec traffic.

To clear IPsec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears the full SA database, which clears active security sessions.)

What to Do Next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see the “[Applying Crypto Map Sets to Interfaces](#)” section.

Creating Crypto Map Entries to Establish Manual SAs

Perform this task to create crypto map entries to establish manual SAs (that is, when IKE is not used to establish the SAs). To create IPv6 crypto maps entries, you must use the **ipv6** keyword with the **crypto map** command. For IPv4 crypto maps, use the **crypto map** command without the **ipv6** keyword.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **crypto map [ipv6] map-name seq-num [ipsec-manual]**
4. **match address access-list-id**
5. **set peer {hostname | ip-address}**
6. **set transform-set transform-set-name**
7. Do one of the following:
 - **set session-key inbound ah spi hex-key-string**
 - **set session-key outbound ah spi hex-key-string**
8. Do one of the following:
 - **set session-key inbound esp spi cipher hex-key-string [authenticator hex-key-string]**
 - **set session-key outbound esp spi cipher hex-key-string [authenticator hex-key-string]**
9. **exit**
10. **exit**
11. **show crypto map [interface interface | tag map-name]**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto map [ipv6] map-name seq-num [ipsec-manual] Example: Device(config)# crypto map mymap 10 ipsec-manual	Specifies the crypto map entry to be created or modified and enters crypto map configuration mode. <ul style="list-style-type: none"> • For IPv4 crypto maps, use the crypto map command without the ipv6 keyword.
Step 4	match address access-list-id Example: Device(config-crypto-m)# match address 102	Names an IPsec access list that determines which traffic should be protected by IPsec and which traffic should not be protected by IPsec in the context of this crypto map entry. <ul style="list-style-type: none"> • The access list can specify only one permit entry when IKE is not used.
Step 5	set peer {hostname ip-address} Example: Device(config-crypto-m)# set peer 10.0.0.5	Specifies the remote IPsec peer. This is the peer to which IPsec protected traffic should be forwarded. <ul style="list-style-type: none"> • Only one peer can be specified when IKE is not used.
Step 6	set transform-set transform-set-name	Specifies which transform set should be used.

	Command or Action	Purpose
	Example: <pre>Device(config-crypto-m)# set transform-set someset</pre>	<ul style="list-style-type: none"> This must be the same transform set that is specified in the remote peer's corresponding crypto map entry. Note Only one transform set can be specified when IKE is not used.
Step 7	Do one of the following: <ul style="list-style-type: none"> set session-key inbound ah spi hex-key-string set session-key outbound ah spi hex-key-string Example: <pre>Device(config-crypto-m)# set session-key inbound ah 256 98765432109876549876543210987654</pre> Example: <pre>Device(config-crypto-m)# set session-key outbound ah 256 fedcbafedcbafedcfedcbafedcbafedc</pre>	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. <ul style="list-style-type: none"> This manually specifies the AH security association to be used with protected traffic.
Step 8	Do one of the following: <ul style="list-style-type: none"> set session-key inbound esp spi cipher hex-key-string [authenticator hex-key-string] set session-key outbound esp spi cipher hex-key-string [authenticator hex-key-string] Example: <pre>Device(config-crypto-m)# set session-key inbound esp 256 cipher 0123456789012345</pre> Example: <pre>Device(config-crypto-m)# set session-key outbound esp 256 cipher abcdefabcdefabcd</pre>	Sets the Encapsulating Security Payload (ESP) Security Parameter Indexes (SPI) and keys to apply to inbound and outbound protected traffic if the specified transform set includes the ESP protocol. <p>Or</p> 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. <ul style="list-style-type: none"> This manually specifies the ESP security association to be used with protected traffic.
Step 9	exit Example: <pre>Device(config-crypto-m)# exit</pre>	Exits crypto map configuration mode and returns to global configuration mode.
Step 10	exit Example: <pre>Device(config)# exit</pre>	Exits global configuration mode.
Step 11	show crypto map [interface interface tag map-name] Example: <pre>Device# show crypto map</pre>	Displays your crypto map configuration.

Troubleshooting Tips

For manually established SAs, you must clear and reinitialize the SAs for the changes to take effect. To clear IPsec SAs, use the **clear crypto sa** command with appropriate parameters. (Omitting all parameters clears the entire SA database, which clears active security sessions.)

What to Do Next

After you have successfully created a crypto map set, you must apply the crypto map set to each interface through which IPsec traffic flows. To complete this task, see the “[Applying Crypto Map Sets to Interfaces](#)” section.

Applying Crypto Map Sets to Interfaces

You must apply a crypto map set to each interface through which IPsec traffic flows. Applying the crypto map set to an interface instructs the device to evaluate 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 the crypto map.

Perform this task to apply a crypto map to an interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type/number*
4. **crypto map** *map-name*
5. **exit**
6. **crypto map** *map-name* **local-address** *interface-id*
7. **exit**
8. **show crypto map** [**interface** *interface*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type/number</i> Example: Device(config)# interface FastEthernet 0/0	Configures an interface and enters interface configuration mode.
Step 4	crypto map <i>map-name</i> Example: Device(config-if)# crypto map mymap	Applies a crypto map set to an interface.
Step 5	exit Example: Device(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.

	Command or Action	Purpose
Step 6	crypto map <i>map-name</i> local-address <i>interface-id</i> Example: Device(config)# crypto map mymap local-address loopback0	(Optional) Permits redundant interfaces to share the same crypto map using the same local identity.
Step 7	exit Example: Device(config)# exit	(Optional) Exits global configuration mode.
Step 8	show crypto map [interface <i>interface</i>] Example: Device# show crypto map	(Optional) Displays your crypto map configuration

Configuration Examples for IPsec VPN

Example: Configuring AES-Based Static Crypto Map

This example shows how a static crypto map is configured and how an AES is defined as the encryption method:

```
crypto isakmp policy 10
 encryption aes 256
 authentication pre-share
 group 14
 lifetime 180
crypto isakmp key cisco123 address 10.0.110.1
!
!
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
 mode transport
!
crypto map aesmap 10 ipsec-isakmp
 set peer 10.0.110.1
 set transform-set aesset
 match address 120
!
!
!
voice call carrier capacity active
!
!
mta receive maximum-recipients 0
!
!
interface FastEthernet0/0
 ip address 10.0.110.2 255.255.255.0
 ip nat outside
 no ip route-cache
 no ip mroute-cache
 duplex auto
 speed auto
```

```

crypto map aesmap
!
interface Serial0/0
no ip address
shutdown
!
interface FastEthernet0/1
ip address 10.0.110.1 255.255.255.0
ip nat inside
no ip route-cache
no ip mroute-cache
duplex auto
speed auto
!
ip nat inside source list 110 interface FastEthernet0/0 overload
ip classless
ip route 0.0.0.0 0.0.0.0 10.5.1.1
ip route 10.0.110.0 255.255.255.0 FastEthernet0/0
ip route 172.18.124.0 255.255.255.0 10.5.1.1
ip route 172.18.125.3 255.255.255.255 10.5.1.1
ip http server
!
!
access-list 110 deny ip 10.0.110.0 0.0.0.255 10.0.110.0 0.0.0.255
access-list 110 permit ip 10.0.110.0 0.0.0.255 any
access-list 120 permit ip 10.0.110.0 0.0.0.255 10.0.110.0 0.0.0.255
!

```

Additional References for Configuring Security for VPNs with IPsec

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IKE, IPsec, and PKI configuration commands	<ul style="list-style-type: none"> • Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IKE configuration	<i>Configuring Internet Key Exchange for IPsec VPNs</i>
Suite-B SHA-2 family (HMAC variant) and Elliptic Curve (EC) key pair configuration	<i>Configuring Internet Key Exchange for IPsec VPNs</i>

Related Topic	Document Title
Suite-B Integrity algorithm type transform configuration	<i>Configuring Internet Key Exchange Version 2 (IKEv2)</i>
Suite-B Elliptic Curve Digital Signature Algorithm (ECDSA) signature (ECDSA-sig) authentication method configuration for IKEv2	<i>Configuring Internet Key Exchange Version 2 (IKEv2)</i>
Suite-B Elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation	<ul style="list-style-type: none"> • <i>Configuring Internet Key Exchange for IPsec VPNs</i> • <i>Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site</i>
Suite-B support for certificate enrollment for a PKI	<i>Configuring Certificate Enrollment for a PKI</i>
Set Up VPN between Cisco ASR 100 Series and Google Cloud Platform	Set Up VPN between Cisco ASR 100 Series and Google Cloud Platform
Recommended cryptographic algorithms	Next Generation Encryption

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
<ul style="list-style-type: none"> • CISCO-IPSEC-FLOW-MONITOR-MIB • CISCO-IPSEC-MIB • CISCO-IPSEC-POLICY-MAP-MIB 	<p>To locate and download MIBs for selected platforms, Cisco IOS software releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p>http://www.cisco.com/go/mibs</p>

RFCs

RFCs	Title
RFC 2401	<i>Security Architecture for the Internet Protocol</i>
RFC 2402	<i>IP Authentication Header</i>
RFC 2403	<i>The Use of HMAC-MD5-96 within ESP and AH</i>
RFC 2404	<i>The Use of HMAC-SHA-1-96 within ESP and AH</i>
RFC 2405	<i>The ESP DES-CBC Cipher Algorithm With Explicit IV</i>

RFCs	Title
RFC 2406	<i>IP Encapsulating Security Payload (ESP)</i>
RFC 2407	<i>The Internet IP Security Domain of Interpretation for ISAKMP</i>
RFC 2408	<i>Internet Security Association and Key Management Protocol (ISAKMP)</i>

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Security for VPNs with IPsec

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 2: Feature Information for Configuring Security for IPsec VPNs

Feature Name	Software Releases	Feature Information
Advanced Encryption Standard	12.2(8)T	This feature adds support for the new encryption standard AES, which is a privacy transform for IPsec and IKE and has been developed to replace DES. The following commands were modified by this feature: crypto ipsec transform-set , encryption (IKE policy) , show crypto ipsec transform-set , show crypto isakmp policy .
DES/3DES/AES VPN Encryption Module (AIM-VPN/EPII, AIM-VPN/HPPII, AIM-VPN/BPII Family)	12.3(7)T	This feature describes in which VPN encryption hardware AIM and NM are supported, in certain Cisco IOS software releases.

Feature Name	Software Releases	Feature Information
IKEv2 Proposal Support	15.1(1)T	<p>An IKEv2 proposal is a set of transforms used in the negotiation of IKEv2 SA as part of the IKE_SA_INIT exchange. An IKEv2 proposal is regarded as complete only when it has at least an encryption algorithm, an integrity algorithm, and a Diffie-Hellman (DH) group configured. If no proposal is configured and attached to an IKEv2 policy, then the default proposal is used in negotiation.</p> <p>The following commands were modified by this feature: crypto ikev2 proposal, encryption (ikev2 proposal), group (ikev2 proposal), integrity (ikev2 proposal), show crypto ikev2 proposal.</p>
IPv6 Support for IPsec and IKEv2	15.1(4)M 15.1(1)SY	<p>This feature allows IPv6 addresses to be added to IPsec and IKEv2 protocols.</p> <p>The following commands were introduced or modified: crypto map (global IPsec), crypto map (isakmp), crypto map (Xauth), ipv6 crypto map.</p>
Option to Disable Volume-based IPsec Lifetime Rekey	15.0(1)M	<p>This feature allows customers to disable the IPsec security association rekey when processing large amounts of data.</p> <p>The following commands were modified by this feature: crypto ipsec security association lifetime, set security-association lifetime.</p>
SEAL Encryption	12.3(7)T	<p>This feature adds support for SEAL encryption in IPsec.</p> <p>The following command was modified by this feature: crypto ipsec transform-set.</p>
Software IPPCP (LZS) with Hardware Encryption	12.2(13)T	<p>This feature allows customers to use LZS software compression with IPsec when a VPN module is in Cisco 2600 and Cisco 3600 series routers.</p>
Suite-B Support in IOS SW Crypto	15.1(2)T	<p>Suite-B adds support for four user interface suites of cryptographic algorithms for use with IKE and IPsec that are described in RFC 4869. Each suite consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm.</p> <p>The following command was modified by this feature: crypto ipsec transform-set.</p>

Glossary

anti-replay—Security service where the receiver can reject old or duplicate packets 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 XE IPsec provides this service whenever it provides the data authentication service, except for manually established SAs (that is, SAs established by configuration and not by IKE).

data authentication—Verification of the integrity and origin of the data. 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—Security service in which the protected data cannot be observed.

data flow—Grouping of traffic, identified by a combination of source address or mask, destination address or mask, IP next protocol field, and source and destination ports, where the protocol and port fields can have the values of **any**. IPsec protection is applied to data flows.

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPsec) that require keys. Before any IPsec traffic can be passed, each router/firewall/host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

IPsec—IP Security. 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. IPsec uses IKE to handle the negotiation of protocols and algorithms based on local policy and to generate the encryption and authentication keys to be used by IPsec. IPsec can 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.

peer—In the context of this module, a “peer” is a router or other device that participates in IPsec.

PFS—perfect forward secrecy. 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.

SA—security association. Description of how two or more entities use security services in the context of a particular security protocol (AH or ESP) to communicate securely on behalf of a particular data flow. The transform and the shared secret keys are used for protecting the traffic.

SPI—security parameter index. A number which, together with a destination IP address and security protocol, uniquely identifies a particular security association. Without IKE, the SPI is manually specified for each security association.

transform—List of operations performed 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 module, “tunnel” is a secure communication path between two peers, such as two routers. It does not refer to using IPsec in tunnel mode.

