



Security for VPNs with IPsec Configuration Guide, Cisco IOS XE Release 3S

Americas Headquarters

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: [www.cisco.com go trademarks](http://www.cisco.com/go/trademarks). Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2018 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1

Configuring Security for VPNs with IPsec 1

Finding Feature Information	1
Prerequisites for Configuring Security for VPNs with IPsec	2
Restrictions for Configuring Security for VPNs with IPsec	2
Information About Configuring Security for VPNs with IPsec	3
Supported Standards	3
Supported Encapsulation	5
IPsec Functionality Overview	5
IKEv1 Transform Sets	5
IKEv2 Transform Sets	5
Transform Sets: A Combination of Security Protocols and Algorithms	6
About Transform Sets	6
Cisco IOS Suite-B Support for IKE and IPsec Cryptographic Algorithms	8
Suite-B Requirements	9
Where to Find Suite-B Configuration Information	9
How to Configure IPsec VPNs	10
Creating Crypto Access Lists	10
What to Do Next	11
Configuring Transform Sets for IKEv1 and IKEv2 Proposals	11
Restrictions	11
Configuring Transform Sets for IKEv1	11
Configuring Transform Sets for IKEv2	13
Creating Crypto Map Sets	15
Creating Static Crypto Maps	15
Creating Dynamic Crypto Maps	18
Creating Crypto Map Entries to Establish Manual SAs	22

Applying Crypto Map Sets to Interfaces	24
Configuration Examples for IPsec VPN	25
Example: Configuring AES-Based Static Crypto Map	25
Additional References for Configuring Security for VPNs with IPsec	27
Feature Information for Configuring Security for VPNs with IPsec	29
Glossary	29

CHAPTER 2**IPsec Virtual Tunnel Interfaces 31**

Finding Feature Information	31
Restrictions for IPsec Virtual Tunnel Interfaces	31
Information About IPsec Virtual Tunnel Interfaces	32
Benefits of Using IPsec Virtual Tunnel Interfaces	32
Static Virtual Tunnel Interfaces	33
Dynamic Virtual Tunnel Interfaces	33
Traffic Encryption with the IPsec Virtual Tunnel Interface	34
Multi-SA Support for Dynamic Virtual Interfaces	35
Dynamic Virtual Tunnel Interface Life Cycle	36
Routing with IPsec Virtual Tunnel Interfaces	36
FlexVPN Mixed Mode Support	36
Auto Tunnel Mode Support in IPsec	37
IPSec Mixed Mode Support for VTI	37
How to Configure IPsec Virtual Tunnel Interfaces	37
Configuring Static IPsec Virtual Tunnel Interfaces	37
Configuring BGP over IPsec Virtual Tunnel Interfaces	39
Configuring Dynamic IPsec Virtual Tunnel Interfaces	41
Configuring Multi-SA Support for Dynamic Virtual Tunnel Interfaces Using IKEv1	42
Configuring IPsec Mixed Mode Support for SVTIs	46
Configuring IPsec Mixed Mode Support for Dynamic VTIs	48
Configuration Examples for IPsec Virtual Tunnel Interfaces	49
Example: Static Virtual Tunnel Interface with IPsec	49
Example: Verifying the Results for the IPsec Static Virtual Tunnel Interface	51
Example: VRF-Aware Static Virtual Tunnel Interface	52
Example: Static Virtual Tunnel Interface with QoS	53
Example: Static Virtual Tunnel Interface with Virtual Firewall	53

Example: Dynamic Virtual Tunnel Interface Easy VPN Server	54
Example: Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Server	55
Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under a Virtual Template	56
Example: VRF-Aware IPsec with Dynamic VTI When VRF Is Configured Under a Virtual Template with the Gateway Option in an IPsec Profile	57
Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile	58
Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile and a Gateway Option in an IPsec Profile	59
Example: VRF-Aware IPsec with a Dynamic VTI When a VRF Is Configured Under Both a Virtual Template and an ISAKMP Profile	59
Example: Dynamic Virtual Tunnel Interface with Virtual Firewall	60
Example: Dynamic Virtual Tunnel Interface with QoS	61
Additional References for IPsec Virtual Tunnel Interface	62
Feature Information for IPsec Virtual Tunnel Interfaces	63

CHAPTER 3**Deleting Crypto Sessions of Revoked Peer Certificates 67**

Finding Feature Information	67
Restrictions for Deleting Crypto Sessions of Revoked Peer Certificates	67
Information About Deleting Crypto Sessions of Revoked Peer Certificates	68
How a Crypto Session is Deleted	68
How to Enable Deletion of Crypto Sessions for Revoked Peer Certificates	68
Enabling Deletion of Crypto Sessions	68
Verifying the Delete Crypto Session Capability for a Revoked Peer Certificate	69
Configuration Examples for Deleting Crypto Sessions of Revoked Peer Certificates	70
Example: Enabling Deletion of Crypto Sessions for an IKE Session	70
Example: Enabling Deletion of Crypto Sessions for an IKEv2 Session	70
Additional References for Deleting Crypto Sessions of Revoked Peers	71
Feature Information for Deleting Crypto Sessions of Revoked Peer Certificates	72

CHAPTER 4**Crypto Conditional Debug Support 73**

Finding Feature Information	73
Prerequisites for Crypto Conditional Debug Support	73
Restrictions for Crypto Conditional Debug Support	73

Information About Crypto Conditional Debug Support	74
Supported Condition Types	74
How to Enable Crypto Conditional Debug Support	75
Enabling Crypto Conditional Debug Messages	75
Performance Considerations	75
Disable Crypto Debug Conditions	75
Enabling Crypto Error Debug Messages	77
debug crypto error CLI	77
Configuration Examples for the Crypto Conditional Debug CLIs	78
Enabling Crypto Conditional Debugging Example	78
Disabling Crypto Conditional Debugging Example	78
Additional References	79
Feature Information for Crypto Conditional Debug Support	80

CHAPTER 5**IPv6 over IPv4 GRE Tunnel Protection 81**

Finding Feature Information	81
Prerequisites for IPv6 over IPv4 GRE Tunnel Protection	81
Restrictions for IPv6 over IPv4 GRE Tunnel Protection	81
Information About IPv6 over IPv4 GRE Tunnel Protection	82
GRE Tunnels with IPsec	82
How to Configure IPv6 over IPv4 GRE Tunnel Protection	83
Configuring IPv6 over IPv4 GRE Encryption Using a Crypto Map	83
Configuring IPv6 over IPv4 GRE Encryption Using Tunnel Protection	87
Configuration Examples for IPv6 over IPv4 GRE Tunnel Protection	90
Example: Configuring IPv6 over IPv4 GRE Encryption Using a Crypto Map	90
Example: Configuring IPv6 over IPv4 GRE Encryption Using Tunnel Protection	91
Additional References	91
Feature Information for IPv6 over IPv4 GRE Tunnel Protection	92

CHAPTER 6**RFC 430x IPsec Support 93**

Finding Feature Information	93
Information About RFC 430x IPsec Support	93
RFC 430x IPsec Support Phase 1	93
RFC 430x IPsec Support Phase 2	94

How to Configure RFC 430x IPsec Support	94
Configuring RFC 430x IPsec Support Globally	94
Configuring RFC 430x IPsec Support Per Crypto Map	95
Configuration Examples for RFC 430x IPsec Support	97
Example: Configuring RFC 430x IPsec Support Globally	97
Example: Configuring RFC 430x IPsec Support Per Crypto Map	98
Additional References for RFC 430x IPsec Support	99
Feature Information for RFC 430x IPsec Support	99



CHAPTER 1

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 25
- [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 29

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 ISR G2 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.
- Crypto maps are not supported on tunnel interface of MFR.

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.

- **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. SHA-2 for ISAKMP is supported in Cisco IOS XE 15.3(3)S and later.
- **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, 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.

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.)



Note You cannot specify AH and ESP algorithms on the same transform set in Cisco IOS XE releases.

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. A transform set combination of AH and ESP is not supported in Cisco IOS XE releases.
ESP Encryption Transform (Pick only one.)	esp-aes	ESP with the 128-bit Advanced Encryption Standard (AES) encryption algorithm.
	esp-gcm esp-gmac	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. 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. The esp-gcm and esp-gmac combinations are not supported on the Cisco ASR 1001 routers with the following ESPs: <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). Note When using DES, 3DES, or GMAC transforms on Cisco ASR 1001-X and Cisco ASR 1002-X routers with ESP-100 or ESP-200, there might be a 30% performance degradation compared to AES and GCM.
	esp-3des	ESP with the 168-bit DES encryption algorithm (3DES or Triple DES). (No longer recommended).
	esp-null	Null encryption algorithm.

Transform Type	Transform	Description
ESP Authentication Transform (Pick only one.)	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.



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.

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. <ul style="list-style-type: none"> • Enter your password if prompted.

What to Do Next

	Command or Action	Purpose
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	crypto ipsec transform-set transform-set-name transform1 [transform2 [transform3]] Example: Device(config)# crypto ipsec transform-set aasset esp-aes 256 esp-sha-hmac	Defines a transform set and enters crypto transform configuration mode. <ul style="list-style-type: none"> 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 [tunnel transport] Example: Device(cfg-crypto-tran)# mode transport	(Optional) Changes the mode associated with the transform set. <ul style="list-style-type: none"> 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: Device(cfg-crypto-tran)# end	Exits crypto transform configuration mode and enters privileged EXEC mode.
Step 6	clear crypto sa [peer {ip-address peer-name} sa map map-name sa entry destination-address protocol spi] 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 [tag transform-set-name] 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.
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.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • 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. • 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. <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-isakmp] Example: Device(config)# crypto map static-map 1 ipsec-isakmp	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: Device(config-crypto-m)# match address vpn-tunnel	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: Device(config-crypto-m)# set-peer 192.168.101.1	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: Device(config-crypto-m)# set security-association dummy seconds 5	Enables generating dummy packets. These dummy packets are generated for all flows created in the crypto map.

	Command or Action	Purpose
Step 7	<p>set transform-set <i>transform-set-name1</i> [<i>transform-set-name2...transform-set-name6</i>]</p> <p>Example: Device(config-crypto-m)# set transform-set aasset</p>	<p>Specifies the transform sets that are allowed for this crypto map entry.</p> <ul style="list-style-type: none"> List multiple transform sets in the order of priority (highest priority first).
Step 8	<p>set security-association lifetime {seconds <i>seconds</i> kilobytes <i>kilobytes</i> kilobytes disable}</p> <p>Example: Device (config-crypto-m)# set security-association lifetime seconds 2700</p>	<p>(Optional) Specifies a SA lifetime for the crypto map entry.</p> <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	<p>set security-association level per-host</p> <p>Example: Device(config-crypto-m)# set security-association level per-host</p>	<p>(Optional) Specifies that separate SAs should be established for each source and destination host pair.</p> <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	<p>set pfs [<i>group1</i> <i>group14</i> <i>group15</i> <i>group16</i> <i>group19</i> <i>group2</i> <i>group20</i> <i>group24</i> <i>group5</i>]</p> <p>Example: Device(config-crypto-m)# set pfs group14</p>	<p>(Optional) Specifies that IPsec either should ask for password forward secrecy (PFS) when requesting new SAs for this crypto map entry or should demand PFS in requests received from the IPsec peer.</p> <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.

	Command or Action	Purpose
Step 11	end Example: Device(config-crypto-m)# end	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: Device# show crypto map	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. <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 dynamic-map <i>dynamic-map-name 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 [transform-set-name2...transform-set-name6]</i> Example: Device(config-crypto-m)# set transform-set aasset	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>

	Command or Action	Purpose
		<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 {hostname ip-address} Example: <pre>Device(config-crypto-m)# set peer 192.168.101.1</pre>	(Optional) Specifies a remote IPsec peer. Repeat this step for multiple remote peers. 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 seconds kilobytes kilobytes kilobytes disable} Example: <pre>Device(config-crypto-m)# set security-association lifetime seconds 7200</pre>	(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: <pre>Device(config-crypto-m)# set pfs group14</pre>	(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.

	Command or Action	Purpose
		<ul style="list-style-type: none"> • 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.
Step 11	show crypto dynamic-map [tag map-name] 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 map-name seq-num ipsec-isakmp dynamic dynamic-map-name [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. Note You must enter the discover keyword to enable TED.
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.

	Command or Action	Purpose
Step 3	<p>crypto map [ipv6] <i>map-name seq-num</i> [ipsec-manual]</p> <p>Example:</p> <pre>Device(config)# crypto map mymap 10 ipsec-manual</pre>	<p>Specifies the crypto map entry to be created or modified and enters crypto map configuration mode.</p> <ul style="list-style-type: none"> For IPv4 crypto maps, use the crypto map command without the ipv6 keyword.
Step 4	<p>match address <i>access-list-id</i></p> <p>Example:</p> <pre>Device(config-crypto-m)# match address 102</pre>	<p>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.</p> <ul style="list-style-type: none"> The access list can specify only one permit entry when IKE is not used.
Step 5	<p>set peer {hostname ip-address}</p> <p>Example:</p> <pre>Device(config-crypto-m)# set peer 10.0.0.5</pre>	<p>Specifies the remote IPsec peer. This is the peer to which IPsec protected traffic should be forwarded.</p> <ul style="list-style-type: none"> Only one peer can be specified when IKE is not used.
Step 6	<p>set transform-set <i>transform-set-name</i></p> <p>Example:</p> <pre>Device(config-crypto-m)# set transform-set someset</pre>	<p>Specifies which transform set should be used.</p> <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. <p>Note Only one transform set can be specified when IKE is not used.</p>
Step 7	<p>Do one of the following:</p> <ul style="list-style-type: none"> set session-key inbound ah <i>spi hex-key-string</i> set session-key outbound ah <i>spi hex-key-string</i> <p>Example:</p> <pre>Device(config-crypto-m)# set session-key inbound ah 256 98765432109876549876543210987654</pre> <p>Example:</p> <pre>Device(config-crypto-m)# set session-key outbound ah 256 fedcbafedcbafedcfedcbafedcbafedc</pre>	<p>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.</p> <ul style="list-style-type: none"> This manually specifies the AH security association to be used with protected traffic.
Step 8	<p>Do one of the following:</p> <ul style="list-style-type: none"> set session-key inbound esp <i>spi cipher hex-key-string</i> [authenticator <i>hex-key-string</i>] set session-key outbound esp <i>spi cipher hex-key-string</i> [authenticator <i>hex-key-string</i>] <p>Example:</p> <pre>Device(config-crypto-m)# set session-key inbound esp 256 cipher 0123456789012345</pre> <p>Example:</p>	<p>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> <p>Or</p> <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.</p> <ul style="list-style-type: none"> This manually specifies the ESP security association to be used with protected traffic.

	Command or Action	Purpose
	Device(config-crypto-m)# set session-key outbound esp 256 cipher abcdefabcdefabcd	
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.
Step 11	show crypto map [interface <i>interface</i> tag <i>map-name</i>] Example: Device# show crypto map	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. <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	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.
Step 6	crypto map <i>map-name local-address 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
```

Example: Configuring AES-Based Static Crypto Map

```

    encryption aes 256
    authentication pre-share
    group 14
    lifetime 180
    crypto isakmp key cisco123 address 10.0.110.1
    !
    !
    crypto ipsec transform-set aasset esp-aes 256 esp-sha-hmac
    mode transport
    !
    crypto map aesmap 10 ipsec-isakmp
    set peer 10.0.110.1
    set transform-set aasset
    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>
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>
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	Cisco IOS XE Release 2.1	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 .
IKE Shared Secret Using AAA Server	Cisco IOS XE Release 2.1	The IKE Shared Secret Using AAA Server feature enables key lookup from a AAA server. In Cisco IOS XE Release 2.1, this feature was implemented on Cisco ASR 1000 Series Aggregation Services Routers.
Suite-B Support in IOS SW Crypto	Cisco IOS XE Release 3.7S	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. The following command was modified by this feature: crypto ipsec transform-set .

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.



CHAPTER 2

IPsec Virtual Tunnel Interfaces

IPsec virtual tunnel interfaces (VTIs) provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify the configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.



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 31](#)
- [Restrictions for IPsec Virtual Tunnel Interfaces, on page 31](#)
- [Information About IPsec Virtual Tunnel Interfaces, on page 32](#)
- [How to Configure IPsec Virtual Tunnel Interfaces, on page 37](#)
- [Configuration Examples for IPsec Virtual Tunnel Interfaces, on page 49](#)
- [Additional References for IPsec Virtual Tunnel Interface, on page 62](#)
- [Feature Information for IPsec Virtual Tunnel Interfaces, on page 63](#)

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.

Restrictions for IPsec Virtual Tunnel Interfaces

Fragmentation

Fragmentation is not supported over IPsec tunnel. You can choose to set the lower MTU on hosts to avoid packet fragments or choose to fragment the packets on any device before it reaches ASR 920.

IPsec Transform Set

The IPsec transform set must be configured in tunnel mode only.

IKE Security Association

The Internet Key Exchange (IKE) security association (SA) is bound to the VTI.

IPsec SA Traffic Selectors

Static VTIs (SVTIs) support only a single IPsec SA that is attached to the VTI interface. The traffic selector for the IPsec SA is always “IP any any.”

IPv4

This feature supports SVTIs that are configured to encapsulate IPv4 packets .

Tunnel Protection

Do not configure the **shared** keyword when using the **tunnel mode ipsec ipv4** command for IPsec IPv4 mode.

Traceroute

The traceroute function with crypto offload on VTIs is not supported.

Information About IPsec Virtual Tunnel Interfaces

The use of IPsec VTIs can simplify the configuration process when you need to provide protection for remote access and it provides an alternative to using generic routing encapsulation (GRE) or Layer 2 Tunneling Protocol (L2TP) tunnels for encapsulation. A benefit of using IPsec VTIs is that the configuration does not require static mapping of IPsec sessions to a physical interface. The IPsec tunnel endpoint is associated with an actual (virtual) interface. Because there is a routable interface at the tunnel endpoint, many common interface capabilities can be applied to the IPsec tunnel.

The IPsec VTI allows for the flexibility of sending and receiving both IP unicast and multicast encrypted traffic on any physical interface, such as in the case of multiple paths. Traffic is encrypted or decrypted when it is forwarded from or to the tunnel interface and is managed by the IP routing table. Using IP routing to forward the traffic to the tunnel interface simplifies the IPsec VPN configuration . Because DVTIs function like any other real interface you can apply quality of service (QoS), firewall, and other security services as soon as the tunnel is active.

The following sections provide details about the IPsec VTI:

Benefits of Using IPsec Virtual Tunnel Interfaces

IPsec VTIs allow you to configure a virtual interface to which you can apply features. Features for clear-text packets are configured on the VTI. Features for encrypted packets are applied on the physical outside interface. When IPsec VTIs are used, you can separate the application of features such as Network Address Translation (NAT), ACLs, and QoS and apply them to clear-text, or encrypted text, or both.

There are two types of VTI interfaces: static VTIs (SVTIs) and dynamic VTIs (DVTIs).

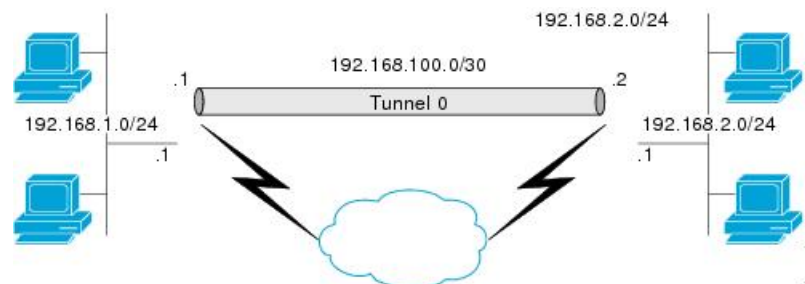
Static Virtual Tunnel Interfaces

SVTI configurations can be used for site-to-site connectivity in which a tunnel provides always-on access between two sites.

Additionally, multiple Cisco IOS software features can be configured directly on the tunnel interface and on the physical egress interface of the tunnel interface. This direct configuration allows users to have solid control on the application of the features in the pre- or post-encryption path.

The figure below illustrates how a SVTI is used.

Figure 1: IPsec SVTI



The IPsec VTI supports native IPsec tunneling and exhibits most of the properties of a physical interface.

Dynamic Virtual Tunnel Interfaces

DVTIs can provide highly secure and scalable connectivity for remote-access VPNs. The DVTI technology replaces dynamic crypto maps and the dynamic hub-and-spoke method for establishing tunnels.



Note You can configure DVTIs with IKEv1 or IKEv2. The legacy crypto map based configuration supports DVTIs with IKEv1 only. A DVTI configuration with IKEv2 is supported only in FlexVPN.

DVTIs can be used for both the server and the remote configuration. The tunnels provide an on-demand separate virtual access interface for each VPN session. The configuration of the virtual access interfaces is cloned from a virtual template configuration, which includes the IPsec configuration and any Cisco IOS software feature configured on the virtual template interface, such as QoS, NetFlow, or ACLs.

DVTIs function like any other real interface, so you can apply QoS, firewall, or other security services as soon as the tunnel is active. QoS features can be used to improve the performance of various applications across the network. Any combination of QoS features offered in Cisco IOS software can be used to support voice, video, or data applications.

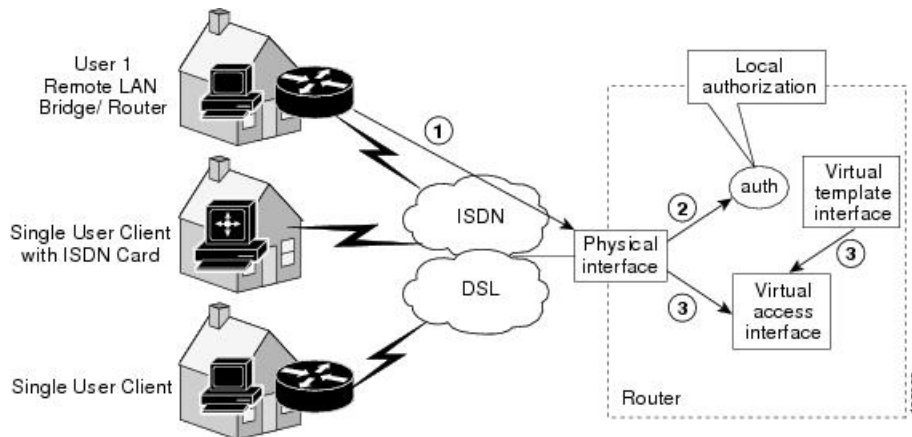
DVTIs provide efficiency in the use of IP addresses and provide secure connectivity. DVTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. The per-group or per-user definition can be created using an extended authentication (Xauth) User or Unity group, or can be derived from a certificate. DVTIs are standards based, so interoperability in a multiple-vendor environment is supported. IPsec DVTIs allow you to create highly secure connectivity for remote access VPNs and can be combined with Cisco Architecture for Voice, Video, and Integrated Data (AVVID) to deliver converged voice, video, and data over IP networks. The DVTI simplifies VPN routing and forwarding- (VRF-) aware IPsec deployment. The VRF is configured on the interface.

A DVTI requires minimal configuration on the router. A single virtual template can be configured and cloned.

The DVTI creates an interface for IPsec sessions and uses the virtual template infrastructure for dynamic instantiation and management of dynamic IPsec VTIs. The virtual template infrastructure is extended to create dynamic virtual-access tunnel interfaces. DVTIs are used in hub-and-spoke configurations. A single DVTI can support several static VTIs.

The figure below illustrates the DVTI authentication path.

Figure 2: Dynamic IPsec VTI



The authentication shown in the figure above follows this path:

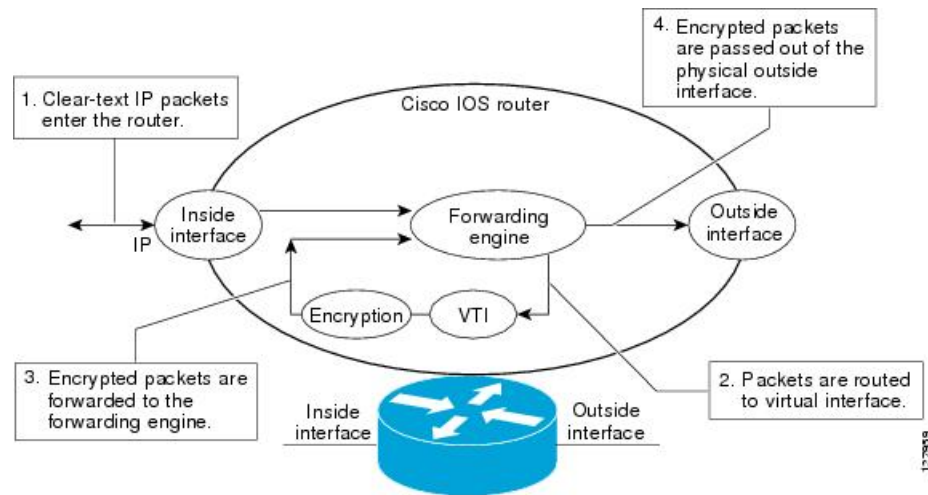
1. User 1 calls the router.
2. Router 1 authenticates User 1.
3. IPsec clones the virtual access interface from the virtual template interface.

Traffic Encryption with the IPsec Virtual Tunnel Interface

When an IPsec VTI is configured, encryption occurs in the tunnel. Traffic is encrypted when it is forwarded to the tunnel interface. Traffic forwarding is handled by the IP routing table, and dynamic or static routing can be used to route traffic to the SVTI. DVTI uses reverse route injection to further simplify the routing configurations. Using IP routing to forward the traffic to encryption simplifies the IPsec VPN configuration. The IPsec virtual tunnel also allows you to encrypt multicast traffic with IPsec.

IPsec packet flow into the IPsec tunnel is illustrated in the figure below.

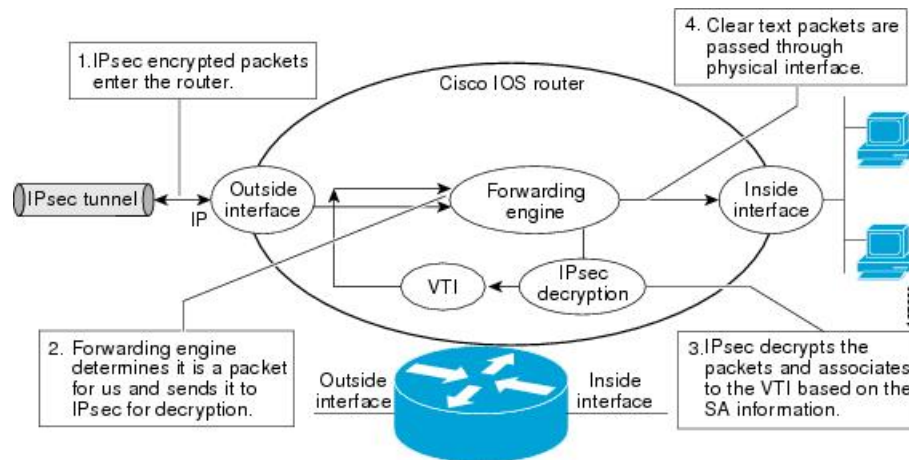
Figure 3: Packet Flow into the IPsec Tunnel



After packets arrive on the inside interface, the forwarding engine switches the packets to the VTI, where they are encrypted. The encrypted packets are handed back to the forwarding engine, where they are switched through the outside interface.

The figure below shows the packet flow out of the IPsec tunnel.

Figure 4: Packet Flow out of the IPsec Tunnel



Multi-SA Support for Dynamic Virtual Interfaces

DVTI supports multiple IPsec SAs. The DVTI can accept multiple IPsec selectors that are proposed by the initiator.

The DVTIs allow per-peer features to be applied on a dedicated interface. You can order features in such way that all features that are applied on the virtual access interfaces are applied before applying crypto. Additionally, all the features that are applied on the physical interfaces are applied after applying crypto. Clean routing is available across all VRFs so that there are no traffic leaks from one VRF to another before encrypting.

Multi-SA VTIs ensure interoperation with third-party devices and provide a flexible, clean, and modular feature-set.

Multi-SA VTIs enable a clean Cisco IOS XE infrastructure, even when the Cisco IOS XE software interoperates with the third-party devices that only implement crypto maps.

VRF and Scalability of Baseline Configuration:

Virtual access instances inherit the Inside-VRF (IVRF) from the template configuration. Users must configure several templates to enforce an appropriate IVRF for each customer. The number of templates must be equal to the number of customers connecting to the headend. Such a configuration is cumbersome and undesirable and also affects performance because each template declaration consumes one Interface Descriptor Block (IDB).

This complication can be avoided by allowing the IKE profile to override the virtual access VRF with the VRF configured on the IKE profile. A better solution is to allow the IKE profile to override the virtual access VRF using AAA, but this method is supported only for IKEv2.

The VRF configured in the ISAKMP profile is applied to the virtual access first. Then the configuration from virtual template is applied to the virtual access. If your virtual template contains **ip vrf forwarding** command configuration, the VRF from the template overrides the VRF from the ISAKMP profile.

Rules for Initial Configuration of a VRF:

The following rules must be applied during the initial configuration of VRF:

- If you configure IVRF in the IKE profile without configuring it in the virtual template, then you must apply the VRF from the IKE profile on each virtual access derived from this IKE profile.
- If you configure VRF in an IKE profile and virtual template, then the virtual template IVRF gets precedence.

Rules for Changing the VRF:

If you change the VRF configured in an IKE profile, all the IKE SAs, IPsec SAs, and the virtual access identifier derived from this profile will get deleted. The same rule applies when the VRF is configured on the IKE profile for the first time.

Dynamic Virtual Tunnel Interface Life Cycle

IPsec profiles define the policy for DVTIs. The dynamic interface is created at the end of IKE Phase 1 and IKE Phase 1.5. The interface is deleted when the IPsec session to the peer is closed. The IPsec session is closed when both IKE and IPsec SAs to the peer are deleted.

Routing with IPsec Virtual Tunnel Interfaces

Because VTIs are routable interfaces, routing plays an important role in the encryption process. Traffic is encrypted only if it is forwarded out of the VTI, and traffic arriving on the VTI is decrypted and routed accordingly. VTIs allow you to establish an encryption tunnel using a real interface as the tunnel endpoint. You can route to the interface or apply services such as QoS, firewalls, network address translation (NAT), and NetFlow statistics as you would to any other interface. You can monitor the interface and route to it, and the interface provides benefits similar to other Cisco IOS interface.

FlexVPN Mixed Mode Support

The FlexVPN Mixed Mode feature provides support for carrying IPv4 traffic over IPsec IPv6 transport. This is the first phase towards providing dual stack support on the IPsec stack. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.

This feature is only supported for Remote Access VPN with IKEv2 and Dynamic VTI.

The FlexVPN Mixed Mode feature provides support for carrying IPv6 traffic over IPsec IPv4 transport from Cisco IOS XE Everest 16.4.1.

Auto Tunnel Mode Support in IPsec

When configuring a VPN headend in a multiple vendor scenario, you must be aware of the technical details of the peer or responder. For example, some devices may use IPsec tunnels while others may use generic routing encapsulation (GRE) or IPsec tunnel, and sometimes, a tunnel may be IPv4 or IPv6. In the last case, you must configure an Internet Key Exchange (IKE) profile and a virtual template.

The Tunnel Mode Auto Selection feature eases the configuration and spares you about knowing the responder's details. This feature automatically applies the tunneling protocol (GRE or IPsec) and transport protocol (IPv4 or IPv6) on the virtual template as soon as the IKE profile creates the virtual access interface. This feature is useful on dual stack hubs aggregating multivendor remote access, such as Cisco AnyConnect VPN Client, Microsoft Windows7 Client, and so on.



Note The Tunnel Mode Auto Selection feature eases the configuration for a responder only. The tunnel must be statically configured for an initiator.

IPsec Mixed Mode Support for VTI

The IPsec Mixed Mode feature provides support for carrying IPv4 traffic over IPsec IPv6 transport. This is the first phase towards providing dual stack support on the IPsec stack. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.

This feature is supported for SVTI as well as DVTI and IKEv1 as well as IKEv2.

How to Configure IPsec Virtual Tunnel Interfaces

Configuring Static IPsec Virtual Tunnel Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto IPsec profile** *profile-name*
4. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
5. **exit**
6. **interface** *type number*
7. **ip address** *address mask*
8. **tunnel mode ipsec ipv4**
9. **tunnel source** *interface-type interface-number*
10. **tunnel destination** *ip-address*

11. tunnel protection IPsec profile *profile-name*
12. end

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 IPsec profile <i>profile-name</i> Example: Device(config)# crypto IPsec profile PROF	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices, and enters IPsec profile configuration mode.
Step 4	set transform-set <i>transform-set-name</i> [<i>transform-set-name2</i>...<i>transform-set-name6</i>] Example: Device(ipsec-profile)# set transform-set tset	Specifies which transform sets can be used .
Step 5	exit Example: Device(ipsec-profile)# exit	Exits IPsec profile configuration mode, and enters global configuration mode.
Step 6	interface <i>type number</i> Example: Device(config)# interface tunnel 0	Specifies the interface on which the tunnel will be configured and enters interface configuration mode.
Step 7	ip address <i>address mask</i> Example: Device(config-if)# ip address 10.1.1.1 255.255.255.0	Specifies the IP address and mask.
Step 8	tunnel mode ipsec ipv4 Example: Device(config-if)# tunnel mode ipsec ipv4	Defines the mode for the tunnel.
Step 9	tunnel source <i>interface-type interface-number</i> Example:	Specifies the tunnel source as a loopback interface.

	Command or Action	Purpose
	Device(config-if)# tunnel source loopback 0	
Step 10	tunnel destination <i>ip-address</i> Example: Device(config-if)# tunnel destination 172.16.1.1	Identifies the IP address of the tunnel destination.
Step 11	tunnel protection IPsec profile <i>profile-name</i> Example: Device(config-if)# tunnel protection IPsec profile PROF	Associates a tunnel interface with an IPsec profile.
Step 12	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring BGP over IPsec Virtual Tunnel Interfaces

Perform this task to optionally configure BGP over the virtual tunnel interfaces of two routers.

Before you begin

Perform steps in [Configuring Static IPsec Virtual Tunnel Interfaces, on page 37](#).

SUMMARY STEPS

1. **router bgp** *autonomous-system-number*
2. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
3. **network** *network-ip-address* **mask** *subnet-mask*
4. **exit**
5. Enter the following commands on the second router.
6. **router bgp** *autonomous-system-number*
7. **neighbor** *ip-address* **remote-as** *autonomous-system-number*
8. **network** *network-ip-address* **mask** *subnet-mask*

DETAILED STEPS

	Command or Action	Purpose
Step 1	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 65510	Enters router configuration mode and creates a BGP routing process. <i>autonomous-system-number</i> —Number of an autonomous system that identifies the router to other BGP routers and tags the routing information that is passed along. Number in the range from 1 to 65535.

	Command or Action	Purpose
		In the example, the first router in this procedure is identified as "65510".
Step 2	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 10.1.1.2 remote-as 65511	<i>ip-address</i> —IP address of the adjacent router's tunnel interface. <i>autonomous-system-number</i> —Number of an autonomous system that identifies the router of the second router. Number in the range from 1 to 65535.
Step 3	network <i>network-ip-address</i> mask <i>subnet-mask</i> Example: Device(config-router)# network 2.2.2.0 mask 255.255.255.0	<i>network-ip-address</i> —IP address of the network advertised in BGP. For example, the IP address of a loopback interface. <i>subnet-mask</i> —subnet mask of the network advertised in BGP. Note The BGP network command network and mask <i>must</i> exactly match a route that is already in the routing table for it to be brought into BGP and advertised to BGP neighbors. This is different from EIGRP, OSPF where the network statement just has to "cover" an interface network and it will pick up the network with mask from the interface.
Step 4	exit Example: Device(config-router)# exit	Exits router configuration mode.
Step 5	Enter the following commands on the second router.	
Step 6	router bgp <i>autonomous-system-number</i> Example: Device(config)# router bgp 65511	Enters router configuration mode and creates a BGP routing process. <i>autonomous-system-number</i> —Number of an autonomous system that identifies the router to other BGP routers and tags the routing information that is passed along. Number in the range from 1 to 65535. In the example, the second router in this procedure is identified as "65511".
Step 7	neighbor <i>ip-address</i> remote-as <i>autonomous-system-number</i> Example: Device(config-router)# neighbor 10.1.1.1 remote-as 65510	<i>ip-address</i> —IP address of the adjacent router's tunnel interface.
Step 8	network <i>network-ip-address</i> mask <i>subnet-mask</i> Example: Device(config-router)# network 1.1.1.0 mask 255.255.255.0	<i>network-ip-address</i> —IP address of the network advertised in BGP. For example, the IP address of a loopback interface. <i>subnet-mask</i> —subnet mask of the network advertised in BGP.

	Command or Action	Purpose
		Note Use the exact network IP address and subnet mask.

Configuring Dynamic IPsec Virtual Tunnel Interfaces

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *profile-name*
4. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
5. **exit**
6. **interface virtual-template** *number* **type tunnel**
7. **tunnel mode ipsec ipv4**
8. **tunnel protection IPsec profile** *profile-name*
9. **exit**
10. **crypto isakamp profile** *profile-name*
11. **match identity address** *ip-address* *mask*
12. **virtual template** *template-number*
13. **end**

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 ipsec profile <i>profile-name</i> Example: Device(config)# crypto ipsec profile PROF	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters IPsec profile configuration mode.
Step 4	set transform-set <i>transform-set-name</i> [<i>transform-set-name2...transform-set-name6</i>] Example: Device(ipsec-profile)# set transform-set tset	Specifies which transform sets can be used with the crypto map entry.
Step 5	exit Example:	Exits ipsec profile configuration mode and enters global configuration mode.

	Command or Action	Purpose
	<code>Device(ipsec-profile)# exit</code>	
Step 6	interface virtual-template <i>number</i> type tunnel Example: <code>Device(config)# interface virtual-template 2 type tunnel</code>	Defines a virtual-template tunnel interface and enters interface configuration mode.
Step 7	tunnel mode ipsec ipv4 Example: <code>Device(config-if)# tunnel mode ipsec ipv4</code>	Defines the mode for the tunnel.
Step 8	tunnel protection IPsec profile <i>profile-name</i> Example: <code>Device(config-if)# tunnel protection ipsec profile PROF</code>	Associates a tunnel interface with an IPsec profile.
Step 9	exit Example: <code>Device(config-if)# exit</code>	Exits interface configuration mode.
Step 10	crypto isakamp profile <i>profile-name</i> Example: <code>Device(config)# crypto isakamp profile profile1</code>	Defines the ISAKMP profile to be used for the virtual template.
Step 11	match identity address <i>ip-address mask</i> Example: <code>Device(conf-isa-prof)# match identity address 10.1.1.0 255.255.255.0</code>	Matches an identity from the ISAKMP profile and enters isakmp-profile configuration mode.
Step 12	virtual template <i>template-number</i> Example: <code>Device(config)# virtual-template 1</code>	Specifies the virtual template attached to the ISAKMP profile.
Step 13	end Example: <code>Device(config)# end</code>	Exits global configuration mode and enters privileged EXEC mode.

Configuring Multi-SA Support for Dynamic Virtual Tunnel Interfaces Using IKEv1



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. **ip vrf** *vrf-name*
4. **rd** *route-distinguisher*
5. **exit**
6. **crypto keyring** *keyring-name*
7. **pre-shared-key** *address key key*
8. **exit**
9. **crypto isakmp profile** *profile-name*
10. **keyring** *keyring-name*
11. **match identity** *address mask*
12. **virtual-template** *template-number*
13. **exit**
14. **crypto ipsec transform-set** *transform-set-name transform1 [transform2] [transform3]*
15. **exit**
16. **crypto ipsec profile** *name*
17. **set security-policy limit** *maximum-limit*
18. **set transform-set** *transform-set-name [transform-set-name2 transform-set-name6]*
19. **exit**
20. **interface virtual-template** *number type tunnel*
21. **ip vrf forwarding** *vrf-name*
22. **ip unnumbered** *type number*
23. **tunnel mode ipsec ipv4**
24. **tunnel protection profile ipsec** *profile-name*
25. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	ip vrf <i>vrf-name</i> Example: Device(config)# ip vrf VRF-100-1	Defines the VRF instance and enters VRF configuration mode.
Step 4	rd <i>route-distinguisher</i> Example: Device(config-vrf)# rd 100:21	Creates routing and forwarding tables for a VRF.

	Command or Action	Purpose
Step 5	exit Example: Device(config-vrf)# exit	Exits VRF configuration mode and enters global configuration mode.
Step 6	crypto keyring <i>keyring-name</i> Example: Device(config)# crypto keyring cisco-100-1	Defines a crypto key ring and enters key ring configuration mode.
Step 7	pre-shared-key <i>address key key</i> Example: Device(config-keyring)# pre-shared-key address 10.1.1.1 key cisco-100-1	Defines the preshared key to be used for Internet Key Exchange (IKE) authentication.
Step 8	exit Example: Device(config-keyring)# exit	Exits keyring configuration mode and enters global configuration mode.
Step 9	crypto isakmp profile <i>profile-name</i> Example: Device(config)# crypto isakmp profile cisco-isakmp-profile-100-1	Defines an ISAKMP profile and enters ISAKMP configuration mode.
Step 10	keyring <i>keyring-name</i> Example: Device(conf-isa-prof)# keyring cisco-100-1	Configures a key ring in ISAKMP mode.
Step 11	match identity <i>address mask</i> Example: Device(conf-isa-prof)# match identity address 10.1.1.0 255.255.255.0	Matches an identity from the ISAKMP profile.
Step 12	virtual-template <i>template-number</i> Example: Device(conf-isa-prof)# virtual-template 101	Specifies the virtual template that will be used to clone virtual access interfaces.
Step 13	exit Example: Device(conf-isa-prof)# exit	Exits ISAKMP profile configuration mode and enters global configuration mode.
Step 14	crypto ipsec transform-set <i>transform-set-name transform1 [transform2] [transform3]</i> Example: Device(config)# crypto ipsec transform-set cisco esp-aes esp-sha-hmac	Defines the transform set and enters crypto transform configuration mode.

	Command or Action	Purpose
Step 15	exit Example: Device(conf-crypto-trans)# exit	Exits crypto transform configuration mode and enters global configuration mode.
Step 16	crypto ipsec profile name Example: Device(config)# crypto ipsec profile cisco-ipsec-profile-101	Defines the IPsec parameters used for IPsec encryption between two IPsec devices, and enters IPsec profile configuration mode.
Step 17	set security-policy limit maximum-limit Example: Device(ipsec-profile)# set security-policy limit 3	Defines an upper limit to the number of flows that can be created for an individual virtual access interface.
Step 18	set transform-set transform-set-name [transform-set-name2 transform-set-name6] Example: Device(ipsec-profile)# set transform-set cisco	Specifies the transform sets to be used with the crypto map entry.
Step 19	exit Example: Device(ipsec-profile)# exit	Exits IPsec profile and enters global configuration mode.
Step 20	interface virtual-template number type tunnel Example: Device(config)# interface virtual-template 101 type tunnel	Creates a virtual template interface that can be configured interface and enters interface configuration mode.
Step 21	ip vrf forwarding vrf-name Example: Device(config-if)# ip vrf forwarding VRF-100-1	Associates a VRF instance with a virtual-template interface.
Step 22	ip unnumbered type number Example: Device(config-if)# ip unnumbered GigabitEthernet 0.0	Enables IP processing on an interface without assigning an explicit IP address to the interface.
Step 23	tunnel mode ipsec ipv4 Example: Device(config-if)# tunnel mode ipsec ipv4	Defines the mode for the tunnel.
Step 24	tunnel protection profile ipsec profile-name Example: Device(config-if)# tunnel protection ipsec profile PROF	Associates a tunnel interface with an IPsec profile.

	Command or Action	Purpose
Step 25	end Example: Device(config-if)# end	Exits interface configuration mode, and returns to privileged EXEC mode.

Configuring IPsec Mixed Mode Support for SVTIs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto IPsec profile** *profile-name*
4. **set transform-set** *transform-set-name* [*transform-set-name2*...*transform-set-name6*]
5. **exit**
6. **interface** *type number*
7. **ip address** *address mask*
8. Do one of the following:
 - **tunnel mode ipsec ipv4 v6-overlay**
 - **tunnel mode ipsec ipv6 v4-overlay**
9. **tunnel source** *interface-type interface-type*
10. **tunnel destination** *ip-address*
11. **tunnel protection IPsec profile** *profile-name*
12. **end**

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 IPsec profile <i>profile-name</i> Example: Device(config)# crypto IPsec profile PROF	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices, and enters IPsec profile configuration mode.

	Command or Action	Purpose
Step 4	set transform-set <i>transform-set-name</i> [<i>transform-set-name2...transform-set-name6</i>] Example: Device(ipsec-profile)# set transform-set tset	Specifies which transform sets can be used with the crypto map entry.
Step 5	exit Example: Device(ipsec-profile)# exit	Exits IPsec profile configuration mode, and enters global configuration mode.
Step 6	interface <i>type number</i> Example: Device(config)# interface tunnel 0	Specifies the interface on which the tunnel will be configured and enters interface configuration mode.
Step 7	ip address <i>address mask</i> Example: Device(config-if)# ip address 10.1.1.1 255.255.255.0	Specifies the IP address and mask.
Step 8	Do one of the following: <ul style="list-style-type: none"> • tunnel mode ipsec ipv4 v6-overlay • tunnel mode ipsec ipv6 v4-overlay Example: Device(config-if)# tunnel mode ipsec ipv4 v6-overlay	Defines the mode for the tunnel.
Step 9	tunnel source <i>interface-type interface-type</i> Example: Device(config-if)# tunnel source loopback 0	Specifies the tunnel source as a loopback interface.
Step 10	tunnel destination <i>ip-address</i> Example: Device(config-if)# tunnel destination 172.16.1.1	Identifies the IP address of the tunnel destination.
Step 11	tunnel protection IPsec profile <i>profile-name</i> Example: Device(config-if)# tunnel protection IPsec profile PROF	Associates a tunnel interface with an IPsec profile.
Step 12	end Example: Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring IPsec Mixed Mode Support for Dynamic VTIs

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *profile-name*
4. **set mixed mode**
5. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
6. **exit**
7. **interface virtual-template** *number* **type tunnel**
8. **tunnel mode ipsec ipv4**
9. **tunnel protection IPsec profile** *profile-name*
10. **exit**
11. **crypto isakamp profile** *profile-name*
12. **match identity address** *ip-address mask*
13. **virtual template** *template-number*
14. **end**

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 profile <i>profile-name</i> Example: Device(config)# crypto ipsec profile PROF	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters IPsec profile configuration mode.
Step 4	set mixed mode Example: Device(config)# set mixed mode	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters IPsec profile configuration mode.
Step 5	set transform-set <i>transform-set-name</i> [<i>transform-set-name2...transform-set-name6</i>] Example: Device(ipsec-profile)# set transform-set tset	Specifies which transform sets can be used with the crypto map entry.
Step 6	exit Example:	Exits ipsec profile configuration mode and enters global configuration mode.

	Command or Action	Purpose
	<code>Device(ipsec-profile)# exit</code>	
Step 7	interface virtual-template <i>number</i> type tunnel Example: <code>Device(config)# interface virtual-template 2 type tunnel</code>	Defines a virtual-template tunnel interface and enters interface configuration mode.
Step 8	tunnel mode ipsec ipv4 Example: <code>Device(config-if)# tunnel mode ipsec ipv4</code>	Defines the mode for the tunnel.
Step 9	tunnel protection IPsec profile <i>profile-name</i> Example: <code>Device(config-if)# tunnel protection ipsec profile PROF</code>	Associates a tunnel interface with an IPsec profile.
Step 10	exit Example: <code>Device(config-if)# exit</code>	Exits interface configuration mode.
Step 11	crypto isakamp profile <i>profile-name</i> Example: <code>Device(config)# crypto isakamp profile profile1</code>	Defines the ISAKMP profile to be used for the virtual template.
Step 12	match identity address <i>ip-address mask</i> Example: <code>Device(conf-isa-prof)# match identity address 10.1.1.0 255.255.255.0</code>	Matches an identity from the ISAKMP profile and enters isakmp-profile configuration mode.
Step 13	virtual template <i>template-number</i> Example: <code>Device(config)# virtual-template 1</code>	Specifies the virtual template attached to the ISAKMP profile.
Step 14	end Example: <code>Device(config)# end</code>	Exits global configuration mode and enters privileged EXEC mode.

Configuration Examples for IPsec Virtual Tunnel Interfaces

Example: Static Virtual Tunnel Interface with IPsec

The following example configuration uses a preshared key for authentication between peers. VPN traffic is forwarded to the IPsec VTI for encryption and then sent out the physical interface. The tunnel on subnet 10

checks packets for the IPsec policy and passes them to the Crypto Engine (CE) for IPsec encapsulation. The figure below illustrates the IPsec VTI configuration.

Figure 5: VTI with IPsec

Router Configuration

```

version 12.3
service timestamps debug datetime
service timestamps log datetime
hostname 7200-3
no aaa new-model
ip subnet-zero
ip cef
controller ISA 6/1
!
crypto isakmp policy 1
encr aes
authentication pre-share
group 14
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set T1 esp-aes esp-sha-hmac
crypto ipsec profile P1
set transform-set T1
!
interface Tunnel0
 ip address 10.0.51.203 255.255.255.0

 load-interval 30
 tunnel source 10.0.149.203
 tunnel destination 10.0.149.217
 tunnel mode IPsec ipv4
 tunnel protection IPsec profile P1
!

 ip address 10.0.149.203 255.255.255.0
 duplex full
!

 ip address 10.0.35.203 255.255.255.0
 duplex full
!
ip classless
ip route 10.0.36.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
end

```

Router Configuration

```

version 12.3
hostname c1750-17
no aaa new-model
ip subnet-zero
ip cef
crypto isakmp policy 1
encr aes
authentication pre-share
group 14
crypto isakmp key Cisco12345 address 0.0.0.0 0.0.0.0

```

```

crypto ipsec transform-set T1 esp-aes esp-sha-hmac
crypto ipsec profile P1
set transform-set T1
!
interface Tunnel0
 ip address 10.0.51.217 255.255.255.0

 tunnel source 10.0.149.217
 tunnel destination 10.0.149.203
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile P1
!
interface
 ip address 10.0.149.217 255.255.255.0
 speed 100
 full-duplex
!
interface
 ip address 10.0.36.217 255.255.255.0
 load-interval 30
 full-duplex
!
ip classless
ip route 10.0.35.0 255.255.255.0 Tunnel0
line con 0
line aux 0
line vty 0 4
end

```

Example: Verifying the Results for the IPsec Static Virtual Tunnel Interface

This section provides information that you can use to confirm that your configuration is working properly. In this display, Tunnel 0 is “up,” and the line protocol is “up.” If the line protocol is “down,” the session is not active.

Verifying the IPsec Static Virtual Tunnel Interface

```

Router# show interface tunnel 0

Tunnel0 is up, line protocol is up
Hardware is Tunnel
Internet address is 10.0.51.203/24
MTU 1514 bytes, BW 9 Kbit, DLY 500000 usec,
reliability 255/255, txload 103/255, rxload 110/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel source 10.0.149.203, destination 10.0.149.217
Tunnel protocol/transport ipsec/ip, key disabled, sequencing disabled
Tunnel TTL 255
Checksumming of packets disabled, fast tunneling enabled
Tunnel transmit bandwidth 8000 (kbps)
Tunnel receive bandwidth 8000 (kbps)
Tunnel protection via IPsec (profile "P1")
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 1/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/0 (size/max)
30 second input rate 13000 bits/sec, 34 packets/sec
30 second output rate 36000 bits/sec, 34 packets/sec
191320 packets input, 30129126 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants, 0 throttles

```

Example: VRF-Aware Static Virtual Tunnel Interface

```
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
59968 packets output, 15369696 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 output buffer failures, 0 output buffers swapped out
```

```
Router# show crypto session
```

```
Crypto session current status
Interface: Tunnel0
Session status: UP-ACTIVE
Peer: 10.0.149.217 port 500
IKE SA: local 10.0.149.203/500 remote 10.0.149.217/500 Active
IPsec FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
Active SAs: 4,
```

```
Router# show ip route
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default, U - per-user static route
o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C 10.0.35.0/24 is directly connected, Ethernet3/3
S 10.0.36.0/24 is directly connected, Tunnel0
C 10.0.51.0/24 is directly connected, Tunnel0
C 10.0.149.0/24 is directly connected, Ethernet3/0
```

Example: VRF-Aware Static Virtual Tunnel Interface

To add the VRF to the static VTI example, include the `ip vrf` and `ip vrf forwarding` commands to the configuration as shown in the following example.

Cisco 7206 Router Configuration

```
hostname cisco 7206
.
.
ip vrf sample-vti1
  rd 1:1
  route-target export 1:1
  route-target import 1:1
!
.
.
interface Tunnel0
  ip vrf forwarding sample-vti1
  ip address 10.0.51.217 255.255.255.0
  tunnel source 10.0.149.217
  tunnel destination 10.0.149.203
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile P1
.
.
!
end
```

Example: Static Virtual Tunnel Interface with QoS

You can apply any QoS policy to the tunnel endpoint by including the **service-policy** statement under the tunnel interface. The following example shows how to police traffic out the tunnel interface.

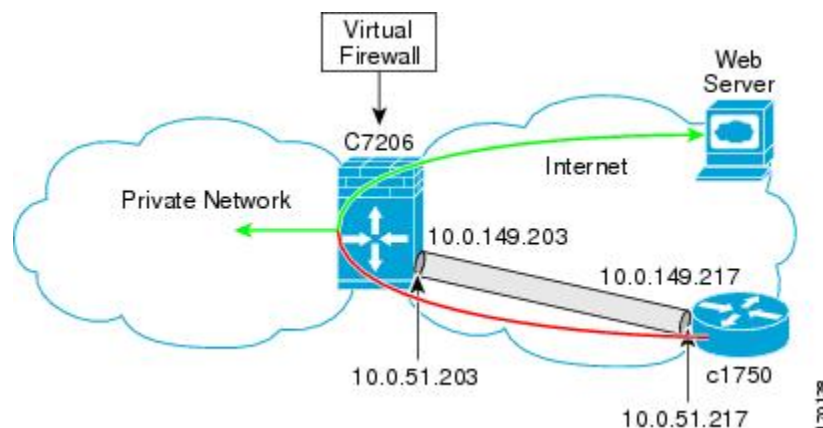
Cisco 7206 Router Configuration

```
hostname cisco 7206
.
.
class-map match-all VTI
  match any
!
policy-map VTI
  class VTI
    police cir 2000000
      conform-action transmit
      exceed-action drop
!
.
.
interface Tunnel0
  ip address 10.0.51.217 255.255.255.0
  tunnel source 10.0.149.217
  tunnel destination 10.0.149.203
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile P1
  service-policy output VTI
!
.
.
!
end
```

Example: Static Virtual Tunnel Interface with Virtual Firewall

Applying the virtual firewall to the SVTI tunnel allows traffic from the spoke to pass through the hub to reach the Internet. The figure below illustrates an SVTI with the spoke protected inherently by the corporate firewall.

Figure 6: Static VTI with Virtual Firewall



The basic SVTI configuration has been modified to include the virtual firewall definition:

Cisco 7206 Router Configuration

```

hostname cisco 7206
.
.
ip inspect max-incomplete high 1000000
ip inspect max-incomplete low 800000
ip inspect one-minute high 1000000
ip inspect one-minute low 800000
ip inspect tcp synwait-time 60
ip inspect tcp max-incomplete host 100000 block-time 2
ip inspect name IOSFW1 tcp timeout 300
ip inspect name IOSFW1 udp
!
.
.
interface GigabitEthernet0/1
description Internet Connection
ip address 172.18.143.246 255.255.255.0
ip access-group 100 in
ip nat outside
!
interface Tunnel0
ip address 10.0.51.217 255.255.255.0
ip nat inside
ip inspect IOSFW1 in
tunnel source 10.0.149.217
tunnel destination 10.0.149.203
tunnel mode ipsec ipv4
tunnel protection ipsec profile P1
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
ip nat translation timeout 120
ip nat translation finrst-timeout 2
ip nat translation max-entries 300000
ip nat pool test1 10.2.100.1 10.2.100.50 netmask 255.255.255.0
ip nat inside source list 110 pool test1 vrf test-vt11 overload
!
access-list 100 permit esp any any
access-list 100 permit udp any eq isakmp any
access-list 100 permit udp any eq non500-isakmp any
access-list 100 permit icmp any any
access-list 110 deny esp any any
access-list 110 deny udp any eq isakmp any
access-list 110 permit ip any any
access-list 110 deny udp any eq non500-isakmp any
!
end

```

Example: Dynamic Virtual Tunnel Interface Easy VPN Server

The following example illustrates the use of the DVTI Easy VPN server, which serves as an IPsec remote access aggregator. The client can be a home user running a Cisco VPN client or a Cisco IOS router configured as an Easy VPN client.

Cisco 7206 Router Configuration

```
hostname cisco 7206
```

```

!
aaa new-model
aaa authentication login local_list local
aaa authorization network local_list local
aaa session-id common
!
ip subnet-zero
ip cef
!
username cisco password 0 cisco123
!
controller ISA 1/1
!
crypto isakmp policy 1
  encr aes
  authentication pre-share
  group 14
!
crypto isakmp client configuration group group1
  key cisco123
  pool group1pool
  save-password
!
crypto isakmp profile vpn1-ra
  match identity group group1
  client authentication list local_list
  isakmp authorization list local_list
  client configuration address respond
  virtual-template 1
!
crypto ipsec transform-set VTI-TS esp-aes esp-sha-hmac
!
crypto ipsec profile test-vt1
  set transform-set VTI-TS
!
interface GigabitEthernet0/1
  description Internet Connection
  ip address 172.18.143.246 255.255.255.0
!
interface GigabitEthernet0/2
  description Internal Network
  ip address 10.2.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
  ip unnumbered GigabitEthernet0/1
  ip virtual-reassembly
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile test-vt1
!
ip local pool group1pool 192.168.1.1 192.168.1.4
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
end

```

Example: Verifying the Results for the Dynamic Virtual Tunnel Interface Easy VPN Server

The following examples show that a DVTI has been configured for an Easy VPN server.

```

Router# show running-config interface Virtual-Access2

Building configuration...
Current configuration : 250 bytes

```

```

!
interface Virtual-Access2
 ip unnumbered GigabitEthernet0/1
 ip virtual-reassembly
 tunnel source 172.18.143.246
 tunnel destination 172.18.143.208
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vtil
 no tunnel protection ipsec initiate
end
Router# show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route
Gateway of last resort is 10.2.1.10 to network 0.0.0.0
 172.18.0.0/24 is subnetted, 1 subnets
C       172.18.143.0 is directly connected, GigabitEthernet0/1
 192.168.1.0/32 is subnetted, 1 subnets
S       192.168.1.1 [1/0] via 0.0.0.0, Virtual-Access2
 10.0.0.0/24 is subnetted, 1 subnets
C       10.2.1.0 is directly connected, GigabitEthernet0/2
S*    0.0.0.0/0 [1/0] via 172.18.143.1

```

Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under a Virtual Template

The following example shows how to configure VRF-aware IPsec under a virtual template to take advantage of the DVTI:

```

hostname cisco 7206
!
ip vrf VRF-100-1
 rd 1:1
!
ip vrf VRF-100-2
 rd 1:1
!
!
!
crypto keyring cisco-100-1
 pre-shared-key address 10.1.1.1 key cisco-100-1
crypto keyring cisco-100-2
 pre-shared-key address 10.1.2.1 key cisco-100-2
crypto isakmp profile cisco-isakmp-profile-100-1
 keyring cisco-100-1
 match identity address 10.1.1.0 255.255.255.0
 virtual-template 101
crypto isakmp profile cisco-isakmp-profile-100-2
 keyring cisco-100-2
 match identity address 10.1.2.0 255.255.255.0
 virtual-template 102
!
!
crypto ipsec transform-set cisco esp-aes esp-sha-hmac
!

```



```

crypto ipsec profile cisco-ipsec-profile-101
  set security-policy limit 3
  set transform-set cisco
!
crypto ipsec profile cisco-ipsec-profile-102
  set security-policy limit 5
  set transform-set Cisco
!
interface Virtual-Template101 type tunnel
  ip vrf forwarding VRF-100-1
  ip unnumbered Ethernet 0/0
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile cisco-ipsec-profile-101
!
interface Virtual-Template102 type tunnel
  ip vrf forwarding VRF-100-2
  ip unnumbered Ethernet 0/0
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile cisco-ipsec-profile-102
!

```

Example: VRF-Aware IPsec with Dynamic VTI When VRF Is Configured Under a Virtual Template with the Gateway Option in an IPsec Profile

The following example shows how to configure VRF-aware IPsec to take advantage of the DVTI, when the VRF is configured under a virtual template with the gateway option in an IPsec profile.

```

hostname ASR 1000
!
ip vrf VRF-100-1
  rd 1:1
!
ip vrf VRF-100-2
  rd 1:1
!
!
!
crypto keyring cisco-100-1
  pre-shared-key address 10.1.1.1 key cisco-100-1
crypto keyring cisco-100-2
  pre-shared-key address 10.1.2.1 key cisco-100-2
crypto isakmp profile cisco-isakmp-profile-100-1
  keyring cisco-100-1
  match identity address 10.1.1.0 255.255.255.0
  virtual-template 101
crypto isakmp profile cisco-isakmp-profile-100-2
  keyring cisco-100-2
  match identity address 10.1.2.0 255.255.255.0
  virtual-template 102
!
!
crypto ipsec transform-set cisco esp-3des esp-sha-hmac
!
crypto ipsec profile cisco-ipsec-profile-101
  set security-policy limit 3
  set transform-set cisco
  set reverse-route gateway 172.16.0.1
!
crypto ipsec profile cisco-ipsec-profile-102
  set security-policy limit 5

```

```

    set transform-set cisco
    set reverse-route gateway 172.16.0.1
    !
interface Virtual-Template101 type tunnel
ip vrf forwarding VRF-100-1
ip unnumbered Ethernet 0/0
tunnel mode ipsec ipv4
tunnel protection ipsec profile cisco-ipsec-profile-101
!
interface Virtual-Template102 type tunnel
ip vrf forwarding VRF-100-2
ip unnumbered Ethernet 0/0
tunnel mode ipsec ipv4
tunnel protection ipsec profile cisco-ipsec-profile-102
!

```

Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile

```

hostname cisco 7206
!
ip vrf VRF-100-1
    rd 1:1
!
ip vrf VRF-100-2
    rd 1:1
!
crypto keyring cisco-100-1
    pre-shared-key address 10.1.1.1 key cisco-100-1
crypto keyring cisco-100-2
    pre-shared-key address 10.1.2.1 key cisco-100-2
crypto isakmp profile cisco-isakmp-profile-100-1
    vrf VRF-100-1
    keyring cisco-100-1
    match identity address 10.1.1.0 255.255.255.0
    virtual-template 1
crypto isakmp profile cisco-isakmp-profile-100-2
    vrf VRF-100-2
    keyring cisco-100-2
    match identity address 10.1.2.0 255.255.255.0
    virtual-template 1
!
!
crypto ipsec transform-set cisco esp-aes esp-sha-hmac
crypto ipsec profile cisco-ipsec-profile
    set security-policy limit 3
    set transform-set cisco
!
!
!
interface Virtual-Template 1 type tunnel
ip unnumbered ethernet 0/0
tunnel mode ipsec ipv4
tunnel protection ipsec profile cisco-ipsec-profile
!
!

```

Example: VRF-Aware IPsec with a Dynamic VTI When VRF Is Configured Under an ISAKMP Profile and a Gateway Option in an IPsec Profile

The following example shows how to configure VRF-aware IPsec to take advantage of the DVTI, when the VRF is configured under an ISAKMP profile and a gateway option in an IPsec profile:

```
hostname ASR 1000
!
ip vrf VRF-100-1
  rd 1:1
!
ip vrf VRF-100-2
  rd 1:1
!
crypto keyring cisco-100-1
  pre-shared-key address 10.1.1.1 key cisco-100-1
crypto keyring cisco-100-2
  pre-shared-key address 10.1.2.1 key cisco-100-2
crypto isakmp profile cisco-isakmp-profile-100-1
  vrf VRF-100-1
  keyring cisco-100-1
  match identity address 10.1.1.0 255.255.255.0
  virtual-template 1
crypto isakmp profile cisco-isakmp-profile-100-2
  vrf VRF-100-2
  keyring cisco-100-2
  match identity address 10.1.2.0 255.255.255.0
  virtual-template 1
!
!
crypto ipsec transform-set cisco esp-3des esp-sha-hmac
crypto ipsec profile cisco-ipsec-profile
  set security-policy limit 3
  set transform-set cisco
  set reverse-route gateway 172.16.0.1
!
!
!
interface Virtual-Templatel type tunnel
  ip unnumbered Ethernet 0/0
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile cisco-ipsec-profile
!
!
```

Example: VRF-Aware IPsec with a Dynamic VTI When a VRF Is Configured Under Both a Virtual Template and an ISAKMP Profile



Note When separate VRFs are configured under an ISAKMP profile and a virtual template, the VRF configured under the virtual template takes precedence. This configuration is not recommended.

The following example shows how to configure VRF-aware IPsec to take advantage of the DVTI when the VRF is configured under both a virtual template and an ISAKMP profile:

```
hostname ASR 1000
.
.
.
ip vrf test-vti2
 rd 1:2
  route-target export 1:1
  route-target import 1:1
!
.
.
.
ip vrf test-vti1
 rd 1:1
  route-target export 1:1
  route-target import 1:1
!
.
.
.
crypto isakmp profile cisco-isakmp-profile
 vrf test-vti2
  keyring key
  match identity address 10.1.1.0 255.255.255.0
!
.
.
.
interface Virtual-Template1 type tunnel
 ip vrf forwarding test-vti1
 ip unnumbered Loopback 0
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vti1
!
.
.
.
end
```

Example: Dynamic Virtual Tunnel Interface with Virtual Firewall

The DVTI Easy VPN server can be configured behind a virtual firewall. Behind-the-firewall configuration allows users to enter the network, while the network firewall is protected from unauthorized access. The virtual firewall uses Context-Based Access Control (CBAC) and NAT applied to the Internet interface as well as to the virtual template.

```
hostname cisco 7206
.
.
ip inspect max-incomplete high 1000000
ip inspect max-incomplete low 800000
ip inspect one-minute high 1000000
ip inspect one-minute low 800000
ip inspect tcp synwait-time 60
ip inspect tcp max-incomplete host 100000 block-time 2
ip inspect name IOSFW1 tcp timeout 300
```

```

ip inspect name IOSFW1 udp
!
.
.
interface GigabitEthernet0/1
  description Internet Connection
  ip address 172.18.143.246 255.255.255.0
  ip access-group 100 in
  ip nat outside
!
interface GigabitEthernet0/2
  description Internal Network
  ip address 10.2.1.1 255.255.255.0
!
interface Virtual-Templatel type tunnel
  ip unnumbered Loopback0
  ip nat inside
  ip inspect IOSFW1 in
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile test-vti1
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.143.1
!
ip nat translation timeout 120
ip nat translation finrst-timeout 2
ip nat translation max-entries 300000
ip nat pool test1 10.2.100.1 10.2.100.50 netmask 255.255.255.0
ip nat inside source list 110 pool test1 vrf test-vti1 overload
!
access-list 100 permit esp any any
access-list 100 permit udp any eq isakmp any
access-list 100 permit udp any eq non500-isakmp any
access-list 100 permit icmp any any
access-list 110 deny esp any any
access-list 110 deny udp any eq isakmp any
access-list 110 permit ip any any
access-list 110 deny udp any eq non500-isakmp any
!
end

```

Example: Dynamic Virtual Tunnel Interface with QoS

You can add QoS to the DVTI tunnel by applying the service policy to the virtual template. When the template is cloned to make the virtual access interface, the service policy will also be applied to the virtual access interface. The following example shows the basic DVTI configuration with QoS added.

```

hostname cisco 7206
.
.
class-map match-all VTI
  match any
!
policy-map VTI
  class VTI
    police cir 2000000
      conform-action transmit
      exceed-action drop
!
.
.

```

```

interface Virtual-Templatel type tunnel
 ip vrf forwarding test-vt1l
 ip unnumbered Loopback0
 ip virtual-reassembly
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile test-vt1l
 service-policy output VTI
!
.
.
!
end

```

Additional References for IPsec Virtual Tunnel Interface

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security 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
IPsec configuration	<i>Configuring Security for VPNs with IPsec</i>
QoS configuration	<i>Cisco IOS Quality of Service Solutions Configuration Guide</i>
EasyVPN configuration	<ul style="list-style-type: none"> • <i>Cisco Easy VPN Remote</i> • <i>Easy VPN Server</i>
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFC 2401	<i>Security Architecture for the Internet Protocol</i>
RFC 2408	<i>Internet Security Association and Key Management Protocol</i>
RFC 2409	<i>The Internet Key Exchange (IKE)</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 IPsec Virtual Tunnel Interfaces

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 3: Feature Information for IPsec Virtual Tunnel Interfaces

Feature Name	Releases	Feature Configuration Information
Dynamic IPsec VTIs	Cisco IOS XE Release 2.1	<p>Dynamic VTIs enable efficient use of IP addresses and provide secure connectivity. Dynamic VTIs allow dynamically downloadable per-group and per-user policies to be configured on a RADIUS server. IPsec dynamic VTIs allow you to create highly secure connectivity for remote access VPNs. The dynamic VTI simplifies VRF-aware IPsec deployment.</p> <p>The following commands were introduced or modified: crypto isakmp profile, interface virtual-template, show vtemplate, tunnel mode, virtual-template.</p>

Feature Name	Releases	Feature Configuration Information
FlexVPN Mixed Mode Support	Cisco IOS XE Release 3.10S	<p>The FlexVPN Mixed Mode feature provides support for carrying IPv4 traffic over IPsec IPv6 transport. This is the first phase towards providing dual stack support on the IPsec stack. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.</p> <p>This feature is only supported for Remote Access VPN with IKEv2 and Dynamic VTI.</p>
Multi-SA for Dynamic VTIs	Cisco IOS XE Release 3.2S	<p>The DVTI can accept multiple IPsec selectors that are proposed by the initiator.</p> <p>The following commands were introduced or modified: set security-policy limit, set reverse-route.</p>
Static IPsec VTIs	Cisco IOS XE Release 2.1	<p>IPsec VTIs provide a routable interface type for terminating IPsec tunnels and an easy way to define protection between sites to form an overlay network. IPsec VTIs simplify configuration of IPsec for protection of remote links, support multicast, and simplify network management and load balancing.</p>
Tunnel Mode Auto Selection	Cisco IOS XE Release 3.12S	<p>The Tunnel Mode Auto Selection feature eases the configuration and spares you about knowing the responder's details. This feature automatically applies the tunneling protocol (GRE or IPsec) and transport protocol (IPv4 or IPv6) on the virtual template as soon as the IKE profile creates the virtual access interface.</p> <p>The following command was introduced or modified: virtual-template</p>

Feature Name	Releases	Feature Configuration Information
FlexVPN Mixed Mode v6 over v4 Transport	Cisco IOS XE Everest 16.4.1	The FlexVPN Mixed Mode v6 over v4 Transport feature provides support for carrying IPv6 traffic over IPsec IPv4 transport. This implementation does not support using a single IPsec security association (SA) pair for both IPv4 and IPv6 traffic.



CHAPTER 3

Deleting Crypto Sessions of Revoked Peer Certificates

The Delete Crypto Sessions of Revoked Peer Certificates on CRL Download feature deletes an active crypto session with a peer if its certificate is found to be revoked when downloading a new CRL.

- [Finding Feature Information, on page 67](#)
- [Restrictions for Deleting Crypto Sessions of Revoked Peer Certificates, on page 67](#)
- [Information About Deleting Crypto Sessions of Revoked Peer Certificates, on page 68](#)
- [How to Enable Deletion of Crypto Sessions for Revoked Peer Certificates, on page 68](#)
- [Configuration Examples for Deleting Crypto Sessions of Revoked Peer Certificates, on page 70](#)
- [Additional References for Deleting Crypto Sessions of Revoked Peers, on page 71](#)
- [Feature Information for Deleting Crypto Sessions of Revoked Peer Certificates, on page 72](#)

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.

Restrictions for Deleting Crypto Sessions of Revoked Peer Certificates

- If revocation check is turned off and this feature is enabled, the IKE database is not populated with the number of sessions. The show outputs do not display information about the deleted sessions.
- Frequent enabling and disabling of this feature (with active sessions on the device) is not recommended.
- Frequent CRL downloads (in a span of 30 minutes) for the same issuername (CA server) is not recommended.

- CRL cache must be enabled. CRL caching cannot be disabled for trustpoint-based prefetch. However, it is possible to disable CRL caching for URL-based prefetch.
- In case of autoenrollment on IKE, the sessions are not deleted until the next IKE rekey, whereas in case of IKEv2, the tunnel must be cleared manually or wait until the certificate expires.
- If IKE has database of “issuer-name” and “SN” populated and receives a notification from PKI about certificate revocation, IKE would act on the PKI notification.

Information About Deleting Crypto Sessions of Revoked Peer Certificates

How a Crypto Session is Deleted

1. When negotiating via certificate authentication, the peer sends the CERT payload to the device, which parses each certificate to store information about serial number and the issuer names. This information forms the list of serial numbers issued by the corresponding CA server and is passed to PKI for revocation check.
2. If the revocation-check crl command is configured for a trustpoint, PKI informs IKE about the revocation check thereby disabling IKE from unnecessarily storing unwanted peer certification information.
3. After a successful CRL download, PKI sends IKE a notification, which contains the “issuer-name.” The CRL signature and content is verified. If there is no change in CRL content, PKI does not notify IKE.
4. If PKI notifies IKE containing the issuer name, IKE prepares a list of serial numbers for an issuer name and passes this list to PKI to verify if the serial numbers in the list are revoked.
5. PKI performs revocation check on the serial number list received from the IKE and checks the list against the downloaded CRL. The revoked serial number list is returned to IKE.
6. On a notification from PKI containing the list of revoked serial numbers, IKE identifies and deletes sessions pertaining to those serial numbers those sessions.

How to Enable Deletion of Crypto Sessions for Revoked Peer Certificates

Enabling Deletion of Crypto Sessions

Perform this task to enable the deletion of crypto sessions for revoked certificates.

SUMMARY STEPS

1. **enable**
2. **clear crypto session**
3. **configure terminal**

4. Do one of the following:
 - **crypto isakmp disconnect-revoked-peers**
 - **crypto ikev2 disconnect-revoked-peers**
5. **end**

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	clear crypto session Example: Device# clear crypto session	(Optional) Deletes IPsec crypto sessions and IKE and security associations. Note Use this command to enable the feature for previously established sessions, else the feature is enabled for new sessions only.
Step 3	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 4	Do one of the following: <ul style="list-style-type: none"> • crypto isakmp disconnect-revoked-peers • crypto ikev2 disconnect-revoked-peers Example: Device(config)# crypto isakmp disconnect-revoked-peers Example: Device(config)# crypto ikev2 disconnect-revoked-peers	Disconnects IKE or IKEv2 crypto sessions with peers having revoked certificates. For this command to take effect, reconnected the existing sessions.
Step 5	end Example: Device(config)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Verifying the Delete Crypto Session Capability for a Revoked Peer Certificate

Perform this task to verify if the delete crypto session capability is displayed in the show output.

SUMMARY STEPS

1. **enable**
2. **show crypto isakmp peers**

3. show crypto ikev2 session detail

DETAILED STEPS

Step 1 enable

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 show crypto isakmp peers

Example:

```
Device# show crypto isakmp peers
```

Displays Internet Security Association and Key Management Protocol (ISAKMP) peer descriptions.

Step 3 show crypto ikev2 session detail

Example:

```
Device# show crypto ikev2 session detail
```

Displays the status of active Internet Key Exchange Version 2 (IKEv2) sessions.

Configuration Examples for Deleting Crypto Sessions of Revoked Peer Certificates

Example: Enabling Deletion of Crypto Sessions for an IKE Session

```
Device> enable
Device# clear crypto session
Device# configure terminal
Device(config)# crypto isakmp disconnect-revoked-peers
Device# show crypto isakmp peers

Peer: 150.1.1.2 Port: 500 Local: 150.1.1.1
Phase1 id: 150.1.1.2
Disconnect Revoked Peer: Enabled
```

Example: Enabling Deletion of Crypto Sessions for an IKEv2 Session

```
Device> enable
Device# clear crypto session
Device# configure terminal
Device(config)# crypto ikev2 disconnect-revoked-peers
Device# show crypto ikev2 session detail
```

```

Session-id:1, Status:UP-ACTIVE, IKE count:1, CHILD count:1
Tunnel-id  Local          Remote          fvrf/ivrf      Status
1          10.0.0.1/500    10.0.0.2/500    (none)/(none)  READY
    Encr: 3DES, Hash: SHA96, DH Grp:2, Auth: PSK
    Life/Remaining/Active Time: 86400/86157/248 sec
    CE id: 0, Session-id: 1, MIB-id: 1
    Status Description: Negotiation done
    Local spi: 750CBE827434A245      Remote spi: 4353FEDBABEBF24C
    Local id: 10.0.0.1                Remote id: 10.0.0.2
    Local req mess id: 0                Remote req mess id: 0
    Local next mess id: 0              Remote next mess id: 2
    Local req queued: 0                Remote req queued: 0
    Local window: 5                    Remote window: 5
    DPD configured for 0 seconds
    NAT-T is not detected
    Disconnect Revoked Peer: Enabled
Child sa: local selector 10.0.0.1/0 - 10.0.0.1/65535
          remote selector 10.0.0.2/0 - 10.0.0.2/65535
          ESP spi in/out: 0x9360A95/0x6C340600
          CPI in/out: 0x9FE5/0xC776
          AH spi in/out: 0x0/0x0
          Encr: AES CBC, keysize: 128, esp_hmac: SHA96
          ah_hmac: Unknown - 0, comp: IPCOMP_LZS, mode tunnel

```

Additional References for Deleting Crypto Sessions of Revoked Peers

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security 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
Configuring IKE	Configuring Internet Key Exchange for IPsec VPNs
Configuring IKEv2	Configuring Internet Key Exchange Version 2 and FlexVPN Site-to-Site
Recommended cryptographic algorithms	Next Generation Encryption

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 Deleting Crypto Sessions of Revoked Peer Certificates

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 4: Feature Information for Deleting Crypto Sessions of Revoked Peer Certificates

Feature Name	Releases	Feature Information
Delete crypto session(s) of revoked peer cert(s) on CRL download	Cisco IOS XE Release 3.13S	<p>The Delete Crypto Sessions of Revoked Peer Certificates on CRL Download feature deletes an active crypto session with a peer if its certificate is found to be revoked when downloading a new CRL.</p> <p>In Cisco IOS XE Release 3.13S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</p> <p>The following commands were introduced or modified: crypto ikev2 disconnect-revoked-peers, crypto isakmp disconnect-revoked-peers, show crypto isakmp peers, show crypto ikev2 session detail.</p>



CHAPTER 4

Crypto Conditional Debug Support

The Crypto Conditional Debug Support feature introduces new debug commands that allow users to debug an IP Security (IPsec) tunnel on the basis of predefined crypto conditions such as the peer IP address, connection-ID of a crypto engine, and security parameter index (SPI). By limiting debug messages to specific IPsec operations and reducing the amount of debug output, users can better troubleshoot a router with a large number of tunnels.

- [Finding Feature Information, on page 73](#)
- [Prerequisites for Crypto Conditional Debug Support, on page 73](#)
- [Restrictions for Crypto Conditional Debug Support, on page 73](#)
- [Information About Crypto Conditional Debug Support, on page 74](#)
- [How to Enable Crypto Conditional Debug Support, on page 75](#)
- [Configuration Examples for the Crypto Conditional Debug CLIs, on page 78](#)
- [Additional References, on page 79](#)
- [Feature Information for Crypto Conditional Debug Support, on page 80](#)

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 Crypto Conditional Debug Support

To use the new crypto CLIs, you must be using the k9 subsystem crypto image in Cisco IOS XE releases.

Restrictions for Crypto Conditional Debug Support

.

- Although conditional debugging is useful for troubleshooting peer-specific or functionality related Internet Key Exchange (IKE) and IPsec problems, conditional debugging may not be able to define and check large numbers of debug conditions. Because extra space is needed to store the debug condition values, additional processing overhead is added to the CPU and memory usage is increased. Thus, enabling crypto conditional debugging on a router with heavy traffic should be used with caution.

Information About Crypto Conditional Debug Support

Supported Condition Types

The new crypto conditional debug CLIs--**debug crypto condition**, **debug crypto condition unmatched**, and **show crypto debug-condition**--allow you to specify conditions (filter values) in which to generate and display debug messages related only to the specified conditions. The table below lists the supported condition types.



Note The **debug crypto condition peer** command with the **ipv4** or **ipv6** keyword can provide the hardware platform specific debugging output. The rest of the condition filters do not provide platform specific debugging output.

Table 5: Supported Condition Types for Crypto Debug CLI

Condition Type (Keyword)	Description
connid ¹	An integer between 1-32766. Relevant debug messages will be shown if the current IPsec operation uses this value as the connection ID to interface with the crypto engine.
FVRF	The name string of a virtual private network (VPN) routing and forwarding (VRF) instance. Relevant debug messages will be shown if the current IPsec operation uses this VRF instance as its front-door VRF (FVRF).
ikev2	The name string for an IKEv2 profile. Relevant debug messages will be shown if the IKEv2 profile name is specified.
isakmp	The name string for an ISAKMP profile. Relevant debug messages will be shown if the ISAKMP profile name is specified.
IVRF	The name string of a VRF instance. Relevant debug messages will be shown if the current IPsec operation uses this VRF instance as its inside VRF (IVRF).
local	The name string of an IPv4 or IPv6 local address.
peer group	A Unity group-name string. Relevant debug messages will be shown if the peer is using this group name as its identity.
peer hostname	A fully qualified domain name (FQDN) string. Relevant debug messages will be shown if the peer is using this string as its identity; for example, if the peer is enabling IKE Xauth with this FQDN string.

Condition Type (Keyword)	Description
peer ipv4 or peer ipv6	A single IP address. Relevant debug messages will be shown if the current IPsec operation is related to the IP address of this peer.
peer subnet	A subnet and a subnet mask that specify a range of peer IP addresses. Relevant debug messages will be shown if the IP address of the current IPsec peer falls into the specified subnet range.
peer username	A username string. Relevant debug messages will be shown if the peer is using this username as its identity; for example, if the peer is enabling IKE Extended Authentication (Xauth) with this username.
session	Provides information about crypto sessions.
SPI	A 32-bit unsigned integer. Relevant debug messages will be shown if the current IPsec operation uses this value as the SPI.
unmatched	Provides debug messages when context information is unavailable.

¹ If an IPsec connid, flowid, or SPI is used as a debug condition, the debug messages for a related IPsec flow are generated. An IPsec flow has two connids, flowids, and SPIs--one inbound and one outbound. Both two connids, flowids, and SPIs can be used as the debug condition that triggers debug messages for the IPsec flow.

How to Enable Crypto Conditional Debug Support

Enabling Crypto Conditional Debug Messages

Performance Considerations

- Before enabling crypto conditional debugging, you must decide what debug condition types (also known as debug filters) and values will be used. The volume of debug messages is dependent on the number of conditions you define.



Note Specifying numerous debug conditions may consume CPU cycles and negatively affect router performance.

- Your router will perform conditional debugging only after at least one of the global crypto debug commands--**debug crypto isakmp**, **debug crypto ipsec**, and **debug crypto engine**--has been enabled. This requirement helps to ensure that the performance of the router will not be impacted when conditional debugging is not being used.

Disable Crypto Debug Conditions

If you choose to disable crypto conditional debugging, you must first disable any crypto global debug CLIs you have issued ; thereafter, you can disable conditional debugging.



Note The **reset** keyword can be used to disable all configured conditions at one time.

SUMMARY STEPS

1. **enable**
2. **debug crypto condition** [*connid integer engine-id integer*] [*flowid integer engine-id integer*] [*fvr string*] [*ivr string*] [*peer [group string] [hostname string] [ipv4 ipaddress] [subnet subnet mask] [username string] [spi integer] [reset]*]
3. **show crypto debug-condition** {[*peer*] [*connid*] [*spi*] [*fvr*] [*ivr*] [*unmatched*]}
4. **debug crypto isakmp**
5. **debug crypto ipsec**
6. **debug crypto engine**
7. **debug crypto condition unmatched** [*isakmp | ipsec | engine*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: <pre>Router> enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	debug crypto condition [<i>connid integer engine-id integer</i>] [<i>flowid integer engine-id integer</i>] [<i>fvr string</i>] [<i>ivr string</i>] [<i>peer [group string] [hostname string] [ipv4 ipaddress] [subnet subnet mask] [username string] [spi integer] [reset]</i>] Example: <pre>Router# debug crypto condition connid 2000 engine-id 1</pre>	Defines conditional debug filters.
Step 3	show crypto debug-condition {[<i>peer</i>] [<i>connid</i>] [<i>spi</i>] [<i>fvr</i>] [<i>ivr</i>] [<i>unmatched</i>]} Example: <pre>Router# show crypto debug-condition spi</pre>	Displays crypto debug conditions that have already been enabled in the router.
Step 4	debug crypto isakmp Example: <pre>Router# debug crypto isakmp</pre>	Enables global IKE debugging.
Step 5	debug crypto ipsec Example:	Enables global IPSec debugging.

	Command or Action	Purpose
	Router# debug crypto ipsec	
Step 6	debug crypto engine Example: Router# debug crypto engine	Enables global crypto engine debugging.
Step 7	debug crypto condition unmatched [isakmp ipsec engine] Example: Router# debug crypto condition unmatched ipsec	(Optional) Displays debug conditional crypto messages when no context information is available to check against debug conditions. If none of the optional keywords are specified, all crypto-related information will be shown.

Enabling Crypto Error Debug Messages

To enable crypto error debug messages, you must perform the following tasks.

debug crypto error CLI

Enabling the **debug crypto error** command displays only error-related debug messages, thereby, allowing you to easily determine why a crypto operation, such as an IKE negotiation, has failed within your system.



Note When enabling this command, ensure that global crypto debug commands are not enabled; otherwise, the global commands will override any possible error-related debug messages.

SUMMARY STEPS

1. **enable**
2. **debug crypto {isakmp | ipsec | engine} error**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	debug crypto {isakmp ipsec engine} error Example: Router# debug crypto ipsec error	Enables only error debugging messages for a crypto area.

Configuration Examples for the Crypto Conditional Debug CLIs

Enabling Crypto Conditional Debugging Example

The following example shows how to display debug messages when the peer IP address is 10.1.1.1, 10.1.1.2, or 10.1.1.3, and when the connection-ID 2000 of crypto engine 0 is used. This example also shows how to enable global debug crypto CLIs and enable the **show crypto debug-condition** command to verify conditional settings.

```
Router#
debug crypto condition connid 2000 engine-id 1
Router#
debug crypto condition peer ipv4 10.1.1.1
Router#
debug crypto condition peer ipv4 10.1.1.2
Router#
debug crypto condition peer ipv4 10.1.1.3
Router#
debug crypto condition unmatched
! Verify crypto conditional settings.
Router#
show crypto debug-condition
Crypto conditional debug currently is turned ON
IKE debug context unmatched flag:ON
IPsec debug context unmatched flag:ON
Crypto Engine debug context unmatched flag:ON
IKE peer IP address filters:
10.1.1.1 10.1.1.2 10.1.1.3
Connection-id filters:[connid:engine_id]2000:1,
! Enable global crypto CLIs to start conditional debugging.
Router#
debug crypto isakmp
Router#
debug crypto ipsec
Router#
debug crypto engine
```

Disabling Crypto Conditional Debugging Example

The following example shows how to disable all crypto conditional settings and verify that those settings have been disabled:

```
Router#
debug crypto condition reset
! Verify that all crypto conditional settings have been disabled.
Router#
show crypto debug-condition
Crypto conditional debug currently is turned OFF
IKE debug context unmatched flag:OFF
IPsec debug context unmatched flag:OFF
Crypto Engine debug context unmatched flag:OFF
```

Additional References

The following sections provide references to the Crypto Conditional Debug Support feature.

Related Documents

Related Topic	Document Title
IPSec and IKE configuration tasks	“ Internet Key Exchange for IPsec VPNs “ module in the <i>Cisco IOS XE Security Configuration Guide: Secure Connectivity</i>
IPSec and IKE commands	<i>Cisco IOS Security Command Reference</i>

Standards

Standards	Title
None	--

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS XE releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	--

Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	http://www.cisco.com/techsupport

Feature Information for Crypto Conditional Debug Support

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 6: Feature Information for Crypto Conditional Debug Support

Feature Name	Releases	Feature Information
Crypto Conditional Debug Support	Cisco IOS XE Release 2.1	The Crypto Conditional Debug Support feature introduces three new command-line interfaces (CLIs) that allow users to debug an IP Security (IPSec) tunnel on the basis of predefined crypto conditions such as the peer IP address, connection-ID of a crypto engine, and security parameter index (SPI).



CHAPTER 5

IPv6 over IPv4 GRE Tunnel Protection

The IPv6 over IPv4 GRE Tunnel Protection feature allows both IPv6 unicast and multicast traffic to pass through a protected generic routing encapsulation (GRE) tunnel.

- [Finding Feature Information, on page 81](#)
- [Prerequisites for IPv6 over IPv4 GRE Tunnel Protection, on page 81](#)
- [Restrictions for IPv6 over IPv4 GRE Tunnel Protection, on page 81](#)
- [Information About IPv6 over IPv4 GRE Tunnel Protection, on page 82](#)
- [How to Configure IPv6 over IPv4 GRE Tunnel Protection, on page 83](#)
- [Configuration Examples for IPv6 over IPv4 GRE Tunnel Protection, on page 90](#)
- [Additional References, on page 91](#)
- [Feature Information for IPv6 over IPv4 GRE Tunnel Protection, on page 92](#)

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 IPv6 over IPv4 GRE Tunnel Protection

- To enable this feature, you must configure IPsec tunnel protection on an IPv4 GRE tunnel.
- To enable IPv6 multicast, you must configure IPv6 multicast routing.

Restrictions for IPv6 over IPv4 GRE Tunnel Protection

The IPv6 over IPv4 GRE Tunnel Protection feature supports IPv6 over IPv4 point-to-point GRE tunnel protection and not IPv6 over IPv4 mGRE tunnel protection.

Information About IPv6 over IPv4 GRE Tunnel Protection

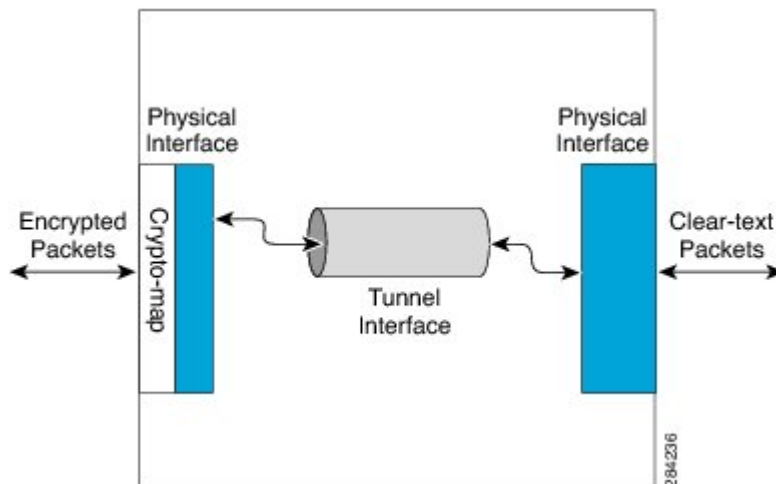
GRE Tunnels with IPsec

Generic routing encapsulation (GRE) tunnels sometimes are combined with IPsec, because IPsec does not support IPv6 multicast packets. This function prevents dynamic routing protocols from running successfully over an IPsec VPN network. Because GRE tunnels do support IPv6 multicast, a dynamic routing protocol can be run over a GRE tunnel. Once a dynamic routing protocol is configured over a GRE tunnel, you can encrypt the GRE IPv6 multicast packets using IPsec.

IPsec can encrypt GRE packets using a crypto map or tunnel protection. Both methods specify that IPsec encryption is performed after GRE encapsulation is configured. When a crypto map is used, encryption is applied to the outbound physical interfaces for the GRE tunnel packets. When tunnel protection is used, encryption is configured on the GRE tunnel interface.

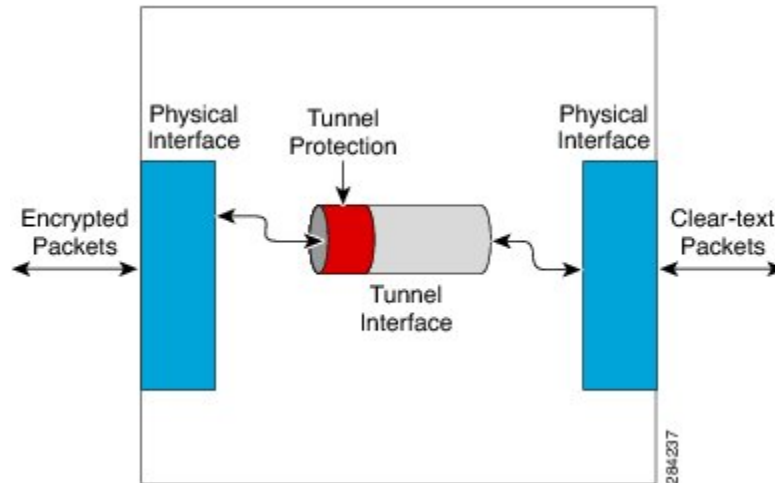
The following figure shows encrypted packets that enter a router through a GRE tunnel interface using a crypto map on the physical interface. Once the packets are decrypted and decapsulated, they continue to their IP destination as clear text.

Figure 7: Using a Crypto Map to Configure IPv6 over IPv4 GRE Tunnel Encryption



The following figure shows encryption using tunnel protection command on the GRE tunnel interface. The encrypted packets enter the router through the tunnel interface and are decrypted and decapsulated before they continue to their destination as clear text.

Figure 8: Using Tunnel Protection to Configure IPv6 over IPv4 GRE Tunnel Encryption



There are two key differences in using the crypto map and tunnel protection methods:

- The IPsec crypto map is tied to the physical interface and is checked as packets are forwarded out through the physical interface. At this point, the GRE tunnel has already encapsulated the packet.
- Tunnel protection ties the encryption functionality to the GRE tunnel and is checked after the packet is GRE encapsulated but before the packet is handed to the physical interface.

How to Configure IPv6 over IPv4 GRE Tunnel Protection

Configuring IPv6 over IPv4 GRE Encryption Using a Crypto Map

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast-routing**
4. **ipv6 unicast-routing**
5. **interface type number**
6. **ipv6 address {ipv6-address/prefix-length | prefix-name sub-bits/prefix-length}**
7. **tunnel mode {aurp | cayman | dvmrp | eon | gre | gre multipoint | gre ip | gre ipv6 | ipip [decapsulate-any] | ipsec ipv4 | iptalk | ipv6 | ipsec ipv6 | mpls | nos | rbscp}**
8. **tunnel source {ip-address | ipv6-address | interface-typeinterface-number}**
9. **tunnel destination {hostname | ip-address | ipv6-address}**
10. **exit**
11. **crypto isakmp policy priority**
12. **authentication {rsa-sig | rsa-encl | pre-share}**
13. **hash {sha | md5}**
14. **group {1 | 2 | 5}**

15. **encryption** {des | 3des | aes 192 | aes 256}
16. **exit**
17. **crypto isakmp key** *enc-type-digit* *keystring* {**address** *peer-address* [*mask*] | **ipv6** {*ipv6-address/ipv6-prefix*} | **hostname** *hostname*} [**no-xauth**]
18. **crypto ipsec transform-set** *transform-set-name* *transform1* [*transform2*] [*transform3*] [*transform4*]
19. **access-list** *access-list-number* [**dynamic** *dynamic-name* [*timeout minutes*]] {**deny** | **permit**} *protocol* *source* *source-wildcard* *destination* *destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**time-range** *time-range-name*] [**fragments**] [**log** [*word*] | **log-input** [*word*]]
20. **crypto map** [**ipv6**] *map-name* *seq-num* [**ipsec-isakmp** [**dynamic** *dynamic-map-name* | **discover** | **profile** *profile-name*]]
21. **set peer** {*hostname* [**dynamic**] [**default**] | *ip-address* [**default**]}
22. **set transform-set** *transform-set-name* [*transform-set-name2*...*transform-set-name6*]
23. **match address** [*access-list-id* | *name*]
24. **exit**
25. **interface** *type number*
26. **crypto map** *map-name* [**redundancy** *standby-group-name* [**stateful**]]
27. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none">• Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 multicast-routing Example: Router(config)# ipv6 multicast-routing	Enables multicast routing using Protocol Independent Multicast (PIM) and Multicast Listener Discovery (MLD) on all IPv6-enabled interfaces of the router and enables multicast forwarding. <ul style="list-style-type: none">• Enable this command only if you are using IPv6 multicast. If you are using IPv6 unicast, you need not enable this command.
Step 4	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.
Step 5	interface <i>type number</i> Example: Router(config)# interface tunnel 10	Specifies a tunnel interface and number, and enters interface configuration mode.

	Command or Action	Purpose
Step 6	ipv6 address { <i>ipv6-address/prefix-length</i> <i>prefix-name sub-bits/prefix-length</i> } Example: Router(config-if)# ipv6 address 0:0:0:7272::72/64	Configures an IPv6 address based on an IPv6 general prefix and enables IPv6 processing on an interface.
Step 7	tunnel mode { <i>aurp</i> <i>cayman</i> <i>dvmrp</i> <i>eon</i> gre gre multipoint gre ip gre ipv6 ipip [<i>decapsulate-any</i>] ipsec ipv4 iptalk ipv6 ipsec ipv6 <i>mpls</i> <i>nos</i> <i>rbscp</i> } Example: Router(config-if)# tunnel mode gre ip	Sets the encapsulation mode for the tunnel interface.
Step 8	tunnel source { <i>ip-address</i> <i>ipv6-address</i> <i>interface-typeinterface-number</i> } Example: Router(config-if)# tunnel source ethernet0	Sets the source address for a tunnel interface.
Step 9	tunnel destination { <i>hostname</i> <i>ip-address</i> <i>ipv6-address</i> } Example: Router(config-if)# tunnel destination 172.16.0.12	Specifies the destination for a tunnel interface.
Step 10	exit Example: Router(config-if)# exit	Exits interface configuration mode and returns to global configuration mode.
Step 11	crypto isakmp policy <i>priority</i> Example: Router(config)# crypto isakmp policy 15	Defines an Internet Key Exchange (IKE) policy, and enters ISAKMP policy configuration mode. <ul style="list-style-type: none"> Policy number 1 indicates the policy with the highest priority. The lower the <i>priority</i> argument value, the higher the priority.
Step 12	authentication { <i>rsa-sig</i> <i>rsa-encr</i> pre-share } Example: Router(config-isakmp-policy)# authentication pre-share	Specifies the authentication method within an IKE policy. <ul style="list-style-type: none"> The <i>rsa-sig</i> and <i>rsa-encr</i> keywords are not supported in IPv6.
Step 13	hash { <i>sha</i> <i>md5</i> } Example: Router(config-isakmp-policy)# hash md5	Specifies the hash algorithm within an IKE policy.
Step 14	group { 1 2 5 } Example: Router(config-isakmp-policy)# group 2	Specifies the Diffie-Hellman group identifier within an IKE policy.
Step 15	encryption { <i>des</i> <i>3des</i> <i>aes 192</i> <i>aes 256</i> } Example:	Specifies the encryption algorithm within an IKE policy.

	Command or Action	Purpose
	Router(config-isakmp-policy)# encryption 3des	
Step 16	exit Example: Router(config-isakmp-policy)# exit	Exits ISAKMP policy configuration mode and enters global configuration mode.
Step 17	crypto isakmp key enc-type-digit keystring {address peer-address [mask] ipv6 {ipv6-address/ipv6-prefix} hostname hostname} [no-xauth] Example: Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0	Configures a preshared authentication key.
Step 18	crypto ipsec transform-set transform-set-name transform1 [transform2] [transform3] [transform4] Example: Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des	Defines a transform set.
Step 19	access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny permit} protocol source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [time-range time-range-name] [fragments] [log [word] log-input [word]] Example: Router(config)# access-list 110 permit gre host 192.168.0.16 host 172.16.0.12	Defines an extended IP access list.
Step 20	crypto map [ipv6] map-name seq-num [ipsec-isakmp [dynamic dynamic-map-name discover profile profile-name]] Example: Router(config)# crypto map mymap 10 ipsec-isakmp	Creates a new crypto map entry or profile and enters crypto map configuration mode.
Step 21	set peer {hostname [dynamic] [default] ip-address [default]} Example: Router(config-crypto-map)# set peer 10.0.0.1	Specifies an IP Security (IPsec) peer in a crypto map entry.
Step 22	set transform-set transform-set-name [transform-set-name2...transform-set-name6] Example: Router(config-crypto-map)# set transform-set myset0	Specifies the transform set that can be used with the crypto map entry.

	Command or Action	Purpose
Step 23	match address [<i>access-list-id</i> <i>name</i>] Example: Router(config-crypto-map)# match address 102	Specifies an extended access list for a crypto map entry.
Step 24	exit Example: Router(config-crypto-map)# exit	Exits crypto map configuration mode and returns to global configuration mode.
Step 25	interface <i>type number</i> Example: Router(config)# interface ethernet 1	Specifies an interface and number and enters interface configuration mode.
Step 26	crypto map <i>map-name</i> [<i>redundancy standby-group-name</i> [<i>stateful</i>]] Example: Router(config-if)# crypto map mymap	Applies a previously defined crypto map set to an outbound interface.
Step 27	end Example: Router(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring IPv6 over IPv4 GRE Encryption Using Tunnel Protection

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 multicast-routing**
4. **ipv6 unicast-routing**
5. **crypto isakmp policy** *priority*
6. **authentication** {*rsa-sig* | *rsa-encr* | *pre-share*}
7. **hash** {*sha* | *md5*}
8. **group** {*1* | *2* | *5*}
9. **encryption** {*des* | *3des* | *aes* | *aes 192* | *aes 256*}
10. **exit**
11. **crypto isakmp key** *enc-type-digit keystring* {*address peer-address* [*mask*] | **ipv6** {*ipv6-address/ipv6-prefix*} | **hostname** *hostname*} [**no-xauth**]
12. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*]
13. **crypto ipsec profile** *profile-name*
14. **set transform-set** *transform-set-name* [*transform-set-name2...transform-set-name6*]
15. **exit**
16. **interface** *type number*
17. **ipv6 address** {*ipv6-address / prefix-length* | *prefix-name sub-bits/prefix-length*}

18. **tunnel mode** {aurp | cayman | dvmrp | eon | gre | gre multipoint | gre ip | gre ipv6 |
 ipip[decapsulate-any] | ipsec ipv4 | iptalk | ipv6 | ipsec ipv6 | mpls | nos | rbsep}
19. **tunnel source** {ip-address | ipv6-address | interface-type interface-number}
20. **tunnel destination** {hostname | ip-address | ipv6-address}
21. **tunnel protection ipsec profile** name [shared]
22. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Router# configure terminal	Enters global configuration mode.
Step 3	ipv6 multicast-routing Example: Router(config)# ipv6 multicast-routing	Enables multicast routing using Protocol Independent Multicast (PIM) and Multicast Listener Discovery (MLD) on all IPv6-enabled interfaces of the router and enables multicast forwarding. <ul style="list-style-type: none"> • Enable this command only if you are using IPv6 multicast. If you are using IPv6 unicast, you do not need to enable this command.
Step 4	ipv6 unicast-routing Example: Router(config)# ipv6 unicast-routing	Enables the forwarding of IPv6 unicast datagrams.
Step 5	crypto isakmp policy priority Example: Router(config)# crypto isakmp policy 15	Defines an IKE policy, and enters ISAKMP policy configuration mode. Policy number 1 indicates the policy with the highest priority. The lower the <i>priority</i> argument value, the higher the priority.
Step 6	authentication {rsa-sig rsa-encr pre-share} Example: Router(config-isakmp-policy)# authentication pre-share	Specifies the authentication method within an Internet Key Exchange (IKE) policy. <ul style="list-style-type: none"> • The rsa-sig and rsa-encr keywords are not supported in IPv6.
Step 7	hash {sha md5} Example: Router(config-isakmp-policy)# hash md5	Specifies the hash algorithm within an IKE policy.

	Command or Action	Purpose
Step 8	group {1 2 5} Example: Router(config-isakmp-policy)# group 2	Specifies the Diffie-Hellman group identifier within an IKE policy.
Step 9	encryption {des 3des aes aes 192 aes 256} Example: Router(config-isakmp-policy)# encryption 3des	Specifies the encryption algorithm within an IKE policy.
Step 10	exit Example: Router(config-isakmp-policy)# exit	Exits ISAKMP policy configuration mode and returns to global configuration mode.
Step 11	crypto isakmp key <i>enc-type-digit</i> <i>keystring</i> { address <i>peer-address</i> [<i>mask</i>] ipv6 { <i>ipv6-address</i> / <i>ipv6-prefix</i> } hostname <i>hostname</i> } [no-xauth] Example: Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0	Configures a preshared authentication key.
Step 12	crypto ipsec transform-set <i>transform-set-name</i> <i>transform1</i> [<i>transform2</i>] [<i>transform3</i>] [<i>transform4</i>] Example: Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des	Defines a transform set, and places the router in crypto transform configuration mode.
Step 13	crypto ipsec profile <i>profile-name</i> Example: Router(config)# crypto ipsec profile ipsecprof	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers and enters IPsec profile configuration mode.
Step 14	set transform-set <i>transform-set-name</i> [<i>transform-set-name2</i> ... <i>transform-set-name6</i>] Example: Router(ipsec-profile)# set transform-set myset0	Specifies the transform set that can be used with the crypto map entry.
Step 15	exit Example: Router(ipsec-profile)# exit	Exits IPsec profile configuration mode and returns to global configuration mode.
Step 16	interface <i>type number</i> Example: Router(config)# interface tunnel 1	Specifies a tunnel interface and number and enters interface configuration mode.
Step 17	ipv6 address { <i>ipv6-address</i> / <i>prefix-length</i> <i>prefix-name</i> <i>sub-bits/prefix-length</i> } Example:	Specifies the IPv6 network assigned to the interface and enables IPv6 processing on the interface.

	Command or Action	Purpose
	Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127	
Step 18	tunnel mode {aurp cayman dvmrp eon gre gre multipoint gre ip gre ipv6 ipip[decapsulate-any] ipsec ipv4 iptalk ipv6 ipsec ipv6 mpls nos rbsep} Example: Router(config-if)# tunnel mode gre ip	Specifies a GRE IPv6 tunnel.
Step 19	tunnel source {ip-address ipv6-address interface-type interface-number} Example: Router(config-if)# tunnel source 10.0.0.1	Specifies the source address or the source interface type and number for the tunnel interface.
Step 20	tunnel destination {hostname ip-address ipv6-address} Example: Router(config-if)# tunnel destination 172.16.0.12	Specifies the destination address or hostname for the tunnel interface.
Step 21	tunnel protection ipsec profile name [shared] Example: Router(config-if)# tunnel protection ipsec profile ipsecprof	Associates a tunnel interface with an IPsec profile.
Step 22	end Example: Router(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

Configuration Examples for IPv6 over IPv4 GRE Tunnel Protection

Example: Configuring IPv6 over IPv4 GRE Encryption Using a Crypto Map

```

Router> enable
Router# configure terminal
Router(config)# ipv6 multicast-routing
Router(config)# ipv6 unicast-routing
Router(config)# interface tunnel 10
Router(config-if)# ipv6 address my-prefix 0:0:0:7272::72/64
Router(config-if)# tunnel mode gre ip
Router(config-if)# tunnel source ethernet0
Router(config-if)# tunnel destination 172.16.0.12
Router(config-if)# exit
Router(config)# crypto isakmp policy 15
Router(config-isakmp-policy)# authentication pre-share
Router(config-isakmp-policy)# hash md5
Router(config-isakmp-policy)# group 2
Router(config-isakmp-policy)# encryption 3des
Router(config-isakmp-policy)# exit
Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0

```

```

Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des
Router(config)# access-list 110 permit gre host 192.168.0.16 host 172.16.0.12
Router(config)# crypto map mymap 10 ipsec-isakmp
Router(config-crypto-map)# set peer 10.0.0.1
Router(config-crypto-map)# set transform-set myset0
Router(config-crypto-map)# match address 102
Router(config-crypto-map)# exit
Router(config)# interface ethernet1
Router(config-if)# crypto map mymap
Router(config-if)# end

```

Example: Configuring IPv6 over IPv4 GRE Encryption Using Tunnel Protection

The following example configures IPsec tunnel protection on an IPv4 GRE tunnel. IPv6 multicast routing is enabled using the `ipv6 multicast-routing` command.

```

Router> enable
Router# configure terminal
Router(config)# ipv6 multicast-routing
Router(config)# ipv6 unicast-routing
Router(config)# crypto isakmp policy 15
Router(config-isakmp-policy)# authentication pre-share
Router(config-isakmp-policy)# hash md5
Router(config-isakmp-policy)# group 2
Router(config-isakmp-policy)# encryption 3des
Router(config-isakmp-policy)# exit
Router(config)# crypto isakmp key cisco-10 address 172.16.0.12 255.240.0.0
Router(config)# crypto ipsec transform-set myset0 ah-sha-hmac esp-3des
Router(config)# crypto ipsec profile ipsecprof
Router(ipsec-profile)# set transform-set myset0
Router(ipsec-profile)# exit
Router(config)# interface tunnel 1
Router(config-if)# ipv6 address 3ffe:b00:c18:1::3/127
Router(config-if)# tunnel mode gre ip
Router(config-if)# tunnel source 10.0.0.1
Router(config-if)# tunnel destination 172.16.0.12
Router(config-if)# tunnel protection ipsec profile ipsecprof
Router(config-if)# end

```

Additional References

Related Documents

Related Topic	Document Title
IPv6 Multicast Routing	IPv6 Implementation Guide
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

Related Topic	Document Title
Security commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<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

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 IPv6 over IPv4 GRE Tunnel Protection

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 7: Feature Information for IPv6 over IPv4 GRE Tunnel Protection

Feature Name	Releases	Feature Information
IPv6 over IPv4 GRE Tunnel Protection	Cisco IOS XE Release 3.5S	<p>The IPv6 over IPv4 GRE tunnel protection feature allows both IPv6 unicast and multicast traffic to pass through a protected GRE tunnel.</p> <p>The following sections provide information about this feature:</p> <ul style="list-style-type: none"> • Information About IPv6 over IPv4 GRE Tunnel Protection • How to Configure IPv6 over IPv4 GRE Tunnel Protection



CHAPTER 6

RFC 430x IPsec Support

The RFC 430x IPsec Support includes features—RFC 430x IPsec Support Phase 1 and RFC430x IPsec Support Phase 2—that implement Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301.

- [Finding Feature Information, on page 93](#)
- [Information About RFC 430x IPsec Support, on page 93](#)
- [How to Configure RFC 430x IPsec Support, on page 94](#)
- [Configuration Examples for RFC 430x IPsec Support, on page 97](#)
- [Additional References for RFC 430x IPsec Support, on page 99](#)
- [Feature Information for RFC 430x IPsec Support, on page 99](#)

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.

Information About RFC 430x IPsec Support

RFC 430x IPsec Support Phase 1

The RFC 430x IPsec Support Phase 1 feature implements Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301.

RFC 4301 specifies the base architecture for IPsec-compliant systems. RFC 4301 describes how to provide a set of security services for traffic at the IP layer, in both the IPv4 and IPv6 environments. The RFC 430x IPsec Support Phase 1 feature provides support for the following RFC 4301 implementations on Cisco IOS software.

- **Security association (SA) lifetime**—The lifetime of a security association between IPsec and Internet Key Exchange (IKE) or Internet Key Exchange Version 2 (IKEv2) must not exceed the lifetime of the authentication certificate.

- **OPAQUE selectors**—OPAQUE indicates that the corresponding selector field is not available for verification. When IKEv2 encounters an OPAQUE selector, IKEv2 skips, does not process the OPAQUE selector, and moves to next selector for policy verification.
- **Explicit Congestion Notification (ECN) support**—ECN is propagated when decrypting an IPsec packet thereby ensuring the packet source and destination are aware of congestion that occurs within the network.
- **Fragment processing**—Peers must not send Initial and noninitial fragments in the same tunnel. There must be a separate tunnel mode SA for carrying initial and noninitial fragments and separate tunnel mode SA for noninitial fragments. IPsec peers must support discarding of packets and stateful fragment checking to accommodate bypass traffic.
- **Do not fragment-(DF) bit processing**—DF-bit processing must be set on a per SA basis.
- **Dummy packet generation support**—It should be possible to send dummy packets via IPsec SA to encapsulate the packets when traffic is flowing via IPsec SA tunnel.

RFC 430x IPsec Support Phase 2

The RFC 430x IPsec Support Phase 2 feature provides support for the RFC 4301 implementation of encryption and decryption of Internet Control Message Protocol (ICMP) packets on Cisco IOS software.

ICMP error messages are sent when an ICMP error occurs. For example, when a host is not reachable, the intermediate device sends a message to the originator of the ICMP request that the host is not reachable. When an ICMP error message reaches an IPsec encryption policy, it may not be classified to match an existing SA. So, the packets are classified based on the data inside the ICMP error message. This data contains the source and destination address of the original ICMP message. If an SA is found based on the address in the ICMP error message, the SA is used. If there is no SA, an SA is created if the policy permits. For decryption, the post decrypt check is performed on the data inside the ICMP error message if a valid SA is not found.

The encryption and decryption of ICMP error messages can be verified through the encrypt and decrypt counters displayed in the output of the **show crypto ipsec sa** command.

Use the conditional debug commands **debug platform condition feature ipsec dataplane submode feature level info**, **debug platform condition both**, and **debug platform condition start** to view ICMP error message classification.

How to Configure RFC 430x IPsec Support

Configuring RFC 430x IPsec Support Globally

Perform this task to configure the RFC 4301 implementations globally.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec security-association dummy {pps rate | seconds seconds}**
4. **crypto ipsec security-association ecn {discard | propogate}**
5. **exit**

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 ipsec security-association dummy {pps rate seconds seconds} Example: Device(config)# crypto ipsec security-association dummy seconds 5	Enables the generation and transmission of dummy packets in an IPsec traffic flow.
Step 4	crypto ipsec security-association ecn {discard propogate} Example: Device(config)# crypto ipsec security-association ecn discard	Enables the Explicit Congestion Notification (ECN) settings in an IPsec traffic flow.
Step 5	exit Example: Device(config-crypto-map)# exit	Exits global configuration mode and returns to privileged EXEC mode.

Configuring RFC 430x IPsec Support Per Crypto Map

Perform this task to configure the RFC 4301 implementations per crypto map.

SUMMARY STEPS

1. enable
2. configure terminal
3. crypto map *map-name seq-num ipsec-isakmp*
4. set ipsec security-association dfbit {clear | copy | set}
5. set ipsec security-association dummy {pps rate | seconds seconds}
6. set ipsec security-association ecn {discard | propogate}
7. end
8. show crypto map ipsec sa

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	<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 <i>map-name seq-num ipsec-isakmp</i> Example: Device(config)# crypto map cmap 1 ipsec-isakmp	Specifies the crypto map entry to be created or modified and enters crypto map configuration mode.
Step 4	set ipsec security-association dfbit {clear copy set} Example: Device(config-crypto-map)# set ipsec security-association dfbit set	Enables do not fragment (DF)-bit processing per security association (SA) for an IPsec traffic flow in a crypto map.
Step 5	set ipsec security-association dummy {pps rate seconds seconds} Example: Device(config-crypto-map)# set ipsec security-association dummy seconds 5	Enables the generation and transmission of dummy packets for an IPsec traffic flow in a crypto map.
Step 6	set ipsec security-association ecn {discard propogate} Example: Device(config-crypto-map)# set ipsec security-association ecn propogate	Enables the Explicit Congestion Notification (ECN) settings per SA for an IPsec traffic flow in a crypto map.
Step 7	end Example: Device(config-crypto-map)# end	Exits crypto map configuration mode and returns to privileged EXEC mode.
Step 8	show crypto map ipsec sa Example: Device# show crypto map ipsec sa	Displays the settings used by IPsec SAs.

Example

The following is sample output from the **show crypto map ipsec sa** command:

```
Device# show crypto map ipsec sa

interface: Tunnel0
  Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::32F7:DFF:FE54:7FD1
  protected vrf: (none)
  local ident (addr/mask/prot/port): (3FFE:2002::32F7:DFF:FE54:7FD1/128/47/0)
  remote ident (addr/mask/prot/port): (3FFE:2002::C671:FEFF:FE88:EB82/128/47/0)
  current_peer 3FFE:2002::C671:FEFF:FE88:EB82 port 500
  PERMIT, flags={origin_is_acl,}
```



```

#pkts encaps: 36, #pkts encrypt: 36, #pkts digest: 36
#pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
#send dummy packets 852600, #recv dummy packets 424905

local crypto endpt.: 3FFE:2002::32F7:DFF:FE54:7FD1,
remote crypto endpt.: 3FFE:2002::C671:FEFF:FE88:EB82
plaintext mtu 1430, path mtu 1500, ipv6 mtu 1500, ipv6 mtu idb GigabitEthernet0/0/1
current outbound spi: 0xE963D1EC(3915633132)
PFS (Y/N): N, DH group: none
Dummy packet: Initializing

inbound esp sas:
spi: 0xF4E01B9A(4108327834)
  transform: esp-3des esp-md5-hmac,
  in use settings =(Tunnel, )
  conn id: 2053, flow_id: ESG:53, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

  sa timing: remaining key lifetime (k/sec): (4608000/2343)
  IV size: 8 bytes
  replay detection support: Y
  Status: ACTIVE(ACTIVE)

inbound ah sas:

inbound pcp sas:

outbound esp sas:
spi: 0xE963D1EC(3915633132)
  transform: esp-3des esp-md5-hmac,
  in use settings =(Tunnel, )
  conn id: 2054, flow_id: ESG:54, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

  sa timing: remaining key lifetime (k/sec): (4608000/2343)
  IV size: 8 bytes
  replay detection support: Y
  Status: ACTIVE(ACTIVE)

outbound ah sas:

outbound pcp sas:

```

Configuration Examples for RFC 430x IPsec Support

Example: Configuring RFC 430x IPsec Support Globally

The following examples shows how to configure RFC 430x IPsec Support globally:

```

Device> enable
Device# configure terminal
Device(config)# crypto ipsec security-association dummy seconds 15
Device(config)# crypto ipsec security-association ecn propogate
Device(config-crypto-map)# exit

```

Example: Configuring RFC 430x IPsec Support Per Crypto Map

The following examples shows how to configure RFC 430x IPsec Support per crypto map:

```

Device> enable
Device# configure terminal
Device(config)# crypto map cmap 1 ipsec-isakmp
Device(config-crypto-map)# set security-association copy
Device(config-crypto-map)# set security-association dummy seconds 15
Device(config-crypto-map)# set security-association ecn propagate
Device(config-crypto-map)# end
Device# show crypto map ipsec sa

interface: Tunnel0
  Crypto map tag: Tunnel0-head-0, local addr 3FFE:2002::32F7:DFF:FE54:7FD1
  protected vrf: (none)
  local ident (addr/mask/prot/port): (3FFE:2002::32F7:DFF:FE54:7FD1/128/47/0)
  remote ident (addr/mask/prot/port): (3FFE:2002::C671:FEFF:FE88:EB82/128/47/0)
  current_peer 3FFE:2002::C671:FEFF:FE88:EB82 port 500
    PERMIT, flags={origin_is_acl,}
  #pkts encaps: 36, #pkts encrypt: 36, #pkts digest: 36
  #pkts decaps: 28, #pkts decrypt: 28, #pkts verify: 28
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0
  #pkts not decompressed: 0, #pkts decompress failed: 0
  #send errors 0, #recv errors 0
  #send dummy packets 852600, #recv dummy packets 424905

  local crypto endpt.: 3FFE:2002::32F7:DFF:FE54:7FD1,
  remote crypto endpt.: 3FFE:2002::C671:FEFF:FE88:EB82
  plaintext mtu 1430, path mtu 1500, ipv6 mtu 1500, ipv6 mtu idb GigabitEthernet0/0/1
  current outbound spi: 0xE963D1EC(3915633132)
  PFS (Y/N): N, DH group: none
  Dummy packet: Initializing

  inbound esp sas:
    spi: 0xF4E01B9A(4108327834)
      transform: esp-3des esp-md5-hmac,
      in use settings = {Tunnel, }
      conn id: 2053, flow_id: ESG:53, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

      sa timing: remaining key lifetime (k/sec): (4608000/2343)
      IV size: 8 bytes
      replay detection support: Y
      Status: ACTIVE(ACTIVE)

  inbound ah sas:

  inbound pcp sas:

  outbound esp sas:
    spi: 0xE963D1EC(3915633132)
      transform: esp-3des esp-md5-hmac,
      in use settings = {Tunnel, }
      conn id: 2054, flow_id: ESG:54, sibling_flags FFFFFFFF80000049, crypto map: Tunnel0-head-0

      sa timing: remaining key lifetime (k/sec): (4608000/2343)
      IV size: 8 bytes
      replay detection support: Y
      Status: ACTIVE(ACTIVE)

  outbound ah sas:

```

```
outbound pcp sas:
```

Additional References for RFC 430x IPsec Support

Related Documents

Related Topic	Document Title
Cisco IOS Commands	Cisco IOS Master Command List, All Releases
Security 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
IKEv2 configuration	Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Site-to-Site
Recommended cryptographic algorithms	Next Generation Encryption

Standards and RFCs

Standard/RFC	Title
RFC 4301	<i>Security Architecture for the Internet Protocol</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 RFC 430x IPsec Support

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 8: Feature Information for RFC430x IPsec Support

Feature Name	Releases	Feature Information
RFC430x IPsec Support Phase 1	Cisco IOS XE Release 3.12S	<p>The RFC 430x IPsec Support Phase 1 feature implements Internet Key Exchange (IKE) and IPsec behavior as specified in RFC 4301.</p> <p>The following commands were introduced or modified: crypto ipsec security-association dummy, crypto ipsec security-association ecn, set ipsec security-association dfbit, set ipsec security-association dummy, set ipsec security-association ecn, show crypto map ipsec sa.</p>
RFC430x IPsec Support Phase 2	Cisco IOS XE Release 3.14S	<p>The RFC 430x IPsec Support Phase 2 feature provides support for the RFC 4301 implementation of encryption and decryption of Internet Control Message Protocol (ICMP) packets on Cisco IOS software.</p> <p>In Cisco IOS XE Release 3.14S, this feature was introduced on Cisco 4431 Integrated Services Router and Cisco 4451-X Integrated Services Router.</p> <p>In Cisco IOS XE Release 3.15S, this feature was implemented on Cisco ASR 1000 Series Aggregation Services Routers and Cisco CSR 1000V Series Cloud Services Router.</p> <p>No commands were modified or updated for this feature.</p>