Configuring IPsec Network Security

IP security (IPsec) protocol is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. It is developed by the Internet Engineering Task Force (IETF). IPsec provides security services at the IP layer, including protecting one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. The overall IPsec implementation is the latest version of RFC 2401. Cisco NX-OS IPsec implements RFC 2402 through RFC 2410.

IPsec uses the Internet Key Exchange (IKE) protocol to handle protocol and algorithm negotiation and to generate the encryption and authentication keys used by IPsec. While IKE can be used with other protocols, its initial implementation is with the IPsec protocol. IKE provides authentication of the IPsec peers, negotiates IPsec security associations, and establishes IPsec keys. IKE uses RFCs 2408, 2409, 2410, and 2412, and additionally implements the draft-ietf-ipsec-ikev2-16.txt draft.

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

This chapter includes the following sections:

Feature Information

This section briefly describes the new and updated features for releases.

Table 7-1 New and Changed Features

Feature	Release	Description
SHA2 support for IPSec and IKEv2 on Cisco MDS 9700 Series Switches	7.3(1)DY(1)	This feature enables SHA2 support for IPSec and IKEv2 on Cisco MDS 9700 Series Switches
SHA2 support for IPSec and IKEv2	7.3(0)D1(1)	This feature enables SHA2 support for IPSec and IKEv2 on a Cisco MDS 9250i Switch.

About IPsec

IP security (IPsec) protocol is a framework of open standards that provides data confidentiality, data integrity, and data authentication between participating peers. It is developed by the Internet Engineering Task Force (IETF). IPsec provides security services at the IP layer, including protecting one or more data flows between a pair of hosts, between a pair of security gateways, or between a security gateway and a host. The overall IPsec implementation is the latest version of RFC 2401. Cisco NX-OS IPsec implements RFC 2402 through RFC 2410.

IPsec uses the Internet Key Exchange (IKE) protocol to handle protocol and algorithm negotiation and to generate the encryption and authentication keys used by IPsec. While IKE can be used with other protocols, its initial implementation is with the IPsec protocol. IKE provides authentication of the IPsec peers, negotiates IPsec security associations, and establishes IPsec keys. IKE uses RFCs 2408, 2409, 2410, and 2412, and additionally implements the draft-ietf-ipsec-ikev2-16.txt draft.

IPsec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers).

Note IPsec is not supported by the Cisco Fabric Switch for HP c-Class BladeSystem and the Cisco Fabric Switch for IBM BladeCenter.

IPsec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPsec acts at the network layer, protecting and authenticating IP packets between participating IPsec devices (peers).

IPsec provides the following network security services. In general, the local security policy dictates the use of one or more of these services between two participating IPsec devices:

- Data confidentiality—The IPsec sender can encrypt packets before transmitting them across a network.
- Data integrity—The IPsec receiver can authenticate packets sent by the IPsec sender to ensure that the data has not been altered during transmission.

- Data origin authentication—The IPsec receiver can authenticate the source of the IPsec packets sent. This service is dependent upon the data integrity service.
- Anti-replay protection—The IPsec receiver can detect and reject replayed packets.

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

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

IPsec as implemented in Cisco NX-OS software supports the Encapsulating Security Payload (ESP) protocol. This protocol encapsulates the data to be protected and provides data privacy services, optional data authentication, and optional anti-replay services.

Note The Encapsulating Security Payload (ESP) protocol is a header inserted into an existing TCP/IP packet, the size of which depends on the actual encryption and authentication algorithms negotiated. To avoid fragmentation, the encrypted packet fits into the interface maximum transmission unit (MTU). The path MTU calculation for TCP takes into account the addition of ESP headers, plus the outer IP header in tunnel mode, for encryption. The MDS switches allow 100 bytes for packet growth for IPsec encryption.

Note When using IPsec and IKE, each Gigabit Ethernet interface on the IPS module (on 18+4, and 24/10 port SAN Extension modules) must be configured in its own IP subnet. If there are multiple Gigabit Ethernet interfaces configured with IP address or network-mask in the same IP subnet, IKE packets may not be sent to the right peer and thus IPsec tunnel will not come up.

About IKE

IKE automatically negotiates IPsec security associations and generates keys for all switches using the IPsec feature. Specifically, IKE provides these benefits:

- Allows you to refresh IPsec SAs.
- Allows IPsec to provide anti-replay services.
- Supports a manageable, scalable IPsec configuration.
- Allows dynamic authentication of peers.

Note IKE is not supported on the Cisco Fabric Switch for HP c-Class BladeSystem and the Cisco Fabric Switch for IBM BladeSystem.

IPsec Prerequisites

To use the IPsec feature, you need to perform the following tasks:

- Obtain the ENTERPRISE_PKG license (see the *Cisco MDS 9000 Family NX-OS Licensing Guide*).

- Configure IKE as described in the "About IKE Initialization" section on page 7-180.

Using IPsec

To use the IPsec feature, follow these steps:

Step 1 Obtain the ENTERPRISE_PKG license to enable IPsec for iSCSI and to enable IPsec for FCIP. See the *Cisco MDS 9000 Family NX-OS Licensing Guide* for more information.

Step 2 Configure IKE as described in the "Manually Configuring IPsec and IKE" section on page 7-180.

Note The IPsec feature inserts new headers in existing packets (see the Cisco MDS 9000 Family NX-OS IP Services Configuration Guide for more information).

This section contains the following topics:

IPsec Compatibility

IPsec features are compatible with the following Cisco MDS 9000 Family hardware:

- Cisco 18/4-port Multi-Service Module (MSM-18/4).
- Cisco MDS 9250i Multiservice Fabric Switches.
- Cisco MDS 24/10 port SAN Extension Module on Cisco MDS 9700 Series Switches.
- The IPsec feature is not supported on the management interface.

IPsec features are compatible with the following fabric setup:

- Two connected Cisco MDS 9200 Switches or Cisco MDS 9500 Directors running Cisco MDS SAN-OS Release 2.0(1b) or later, or Cisco NX-OS 4.1(1).
- A Cisco MDS 9200 Switches or Cisco MDS 9500 Directors running Cisco MDS SAN-OS Release 2.0(1b) or later, or Cisco NX-OS 4.1(1) connected to any IPsec compliant device.
- The following features are not supported in the Cisco NX-OS implementation of the IPsec feature:
  - Authentication Header (AH).
  - Transport mode.
  - Security association bundling.
  - Manually configuring security associations.
  - Per host security association option in a crypto map.
  - Security association idle timeout

- Dynamic crypto maps.

Note Any reference to crypto maps in this document, only refers to static crypto maps.

IPsec and IKE Terminology

The terms used in this chapter are explained in this section.

- Security association (SA)— An agreement between two participating peers on the entries required to encrypt and decrypt IP packets. Two SAs are required for each peer in each direction (inbound and outbound) to establish bidirectional communication between the peers. Sets of bidirectional SA records are stored in the SA database (SAD). IPsec uses IKE to negotiate and bring up SAs. Each SA record includes the following information:
  - Security parameter index (SPI)—A number which, together with a destination IP address and security protocol, uniquely identifies a particular SA. When using IKE to establish the SAs, the SPI for each SA is a pseudo-randomly derived number.
  - Peer—A switch or other device that participates in IPsec. For example, a Cisco MDS switch or other Cisco routers that support IPsec.
  - Transform—A list of operations done to provide data authentication and data confidentiality. For example, one transform is the ESP protocol with the HMAC-MD5 authentication algorithm.
  - Session key—The key used by the transform to provide security services.
  - Lifetime—A lifetime counter (in seconds and bytes) is maintained from the time the SA is created. When the time limit expires the SA is no longer operational and, if required, is automatically renegotiated (rekeyed).
  - Mode of operation—Two modes of operation are generally available for IPsec: tunnel mode and transport mode. The Cisco NX-OS implementation of IPsec only supports the tunnel mode. The IPsec tunnel mode encrypts and authenticates the IP packet, including its header. The gateways encrypt traffic on behalf of the hosts and subnets.
    The Cisco NX-OS implementation of IPsec does not support transport mode.

  Note The term *tunnel mode* is different from the term *tunnel*, which is used to indicate a secure communication path between two peers, such as two switches connected by an FCIP link.

- Anti-replay—A 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.
- Data authentication—Data authentication can refer either to integrity alone or to both integrity and authentication (data origin authentication is dependent on data integrity).
  - Data integrity—Verifies that data has not been altered.
  - Data origin authentication—Verifies that the data was actually sent by the claimed sender.
- Data confidentiality—A security service where the protected data cannot be observed.
- Data flow—A grouping of traffic, identified by a combination of source address and mask or prefix, destination address mask or prefix length, IP next protocol field, and source and destination ports, where the protocol and port fields can have any of these values. Traffic matching a specific

combination of these values is logically grouped together into a data flow. A data flow can represent a single TCP connection between two hosts, or it can represent traffic between two subnets. IPsec protection is applied to data flows.

- Perfect forward secrecy (PFS)—A cryptographic characteristic associated with a derived shared secret value. With PFS, if one key is compromised, previous and subsequent keys are not compromised, because subsequent keys are not derived from previous keys.
- Security Policy Database (SPD)—An ordered list of policies applied to traffic. A policy decides if a packet requires IPsec processing, if it should be allowed in clear text, or if it should be dropped.
  - The IPsec SPDs are derived from user configuration of crypto maps.
  - The IKE SPD is configured by the user.

Supported IPsec Transforms and Algorithms

The component technologies implemented for IPsec include the following transforms:

- Advanced Encrypted Standard (AES) is an encryption algorithm. It implements either 128 or 256 bits using Cipher Block Chaining (CBC) or counter mode.
- Data Encryption Standard (DES) is used to encrypt packet data and implements the mandatory 56-bit DES-CBC. CBC requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPsec packet.
- Triple DES (3DES) is a stronger form of DES with 168-bit encryption keys that allow sensitive information to be transmitted over untrusted networks.

Note Cisco NX-OS images with strong encryption are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- Message Digest 5 (MD5) is a hash algorithm with the HMAC variant. HMAC is a keyed hash variant used to authenticate data.
- Secure Hash Algorithm (SHA-1, SHA-2) is a hash algorithm with the Hash Message Authentication Code (HMAC) variant. IPsec supports SHA-2 on Cisco MDS 9250i Multiservice Fabric Switches starting from Cisco MDS NX-OS Release 7.3(0)D1(1).
- AES-XCBC-MAC is a Message Authentication Code (MAC) using the AES algorithm.
- IPsec supports SHA-2 on Cisco MDS 24/10 port SAN Extension Modules (Cisco MDS 9700 Series Switches) starting from Cisco MDS NX-OS Release 7.3(1)DY(1).

Supported IKE Transforms and Algorithms

The component technologies implemented for IKE include the following transforms:

- Diffie-Hellman (DH) is 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. Group 1 (768-bit), Group 2 (1024-bit), and Group 5 (1536-bit) are supported.

- Advanced Encrypted Standard (AES) is an encryption algorithm. It implements either 128 bits using Cipher Block Chaining (CBC) or counter mode.
- Data Encryption Standard (DES) is used to encrypt packet data and implements the mandatory 56-bit DES-CBC. CBC requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPsec packet.
- Triple DES (3DES) is a stronger form of DES with 168-bit encryption keys that allow sensitive information to be transmitted over untrusted networks.

Note Cisco NX-OS images with strong encryption are subject to United States government export controls, and have a limited distribution. Images to be installed outside the United States require an export license. Customer orders might be denied or subject to delay due to United States government regulations. Contact your sales representative or distributor for more information, or send e-mail to export@cisco.com.

- Message Digest 5 (MD5) is a hash algorithm with the HMAC variant. HMAC is a keyed hash variant used to authenticate data.
- Secure Hash Algorithm (SHA-1, SHA-2) is a hash algorithm with the Hash Message Authentication Code (HMAC) variant. IKEv2 supports SHA-2 on Cisco MDS 9250i Multiservice Fabric Switches starting from Cisco MDS NX-OS Release 7.3(0)D1(1).

Note IKEv1 does not support SHA-2.

- The switch authentication algorithm uses the preshared keys based on the IP address.
- IKEv2 supports SHA-2 on Cisco MDS 24/10 port SAN Extension Modules (Cisco MDS 9700 Series Switches) starting from Cisco MDS NX-OS Release 7.3(1)DY(1).

IPsec Digital Certificate Support

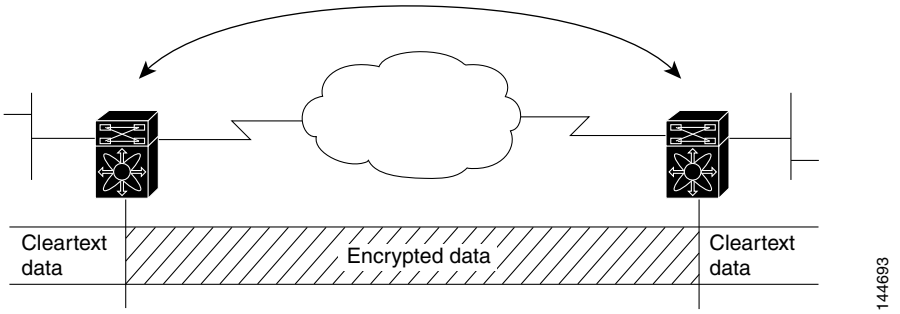
This section describes the advantages of using certificate authorities (CAs) and digital certificates for authentication.

Implementing IPsec Without CAs and Digital Certificates

Without a CA and digital certificates, enabling IPsec services (such as encryption) between two Cisco MDS switches requires that each switch has the key of the other switch (such as an RSA public key or a shared key). You must manually specify either the RSA public keys or preshared keys on each switch in the fabric using IPsec services. Also, each new device added to the fabric will require manual configuration of the other switches in the fabric to support secure communication. Each (see Figure 7-1) switch uses the key of the other switch to authenticate the identity of the other switch; this authentication always occurs when IPsec traffic is exchanged between the two switches.

If you have multiple Cisco MDS switches in a mesh topology and wish to exchange IPsec traffic passing among all of those switches, you must first configure shared keys or RSA public keys among all of those switches.

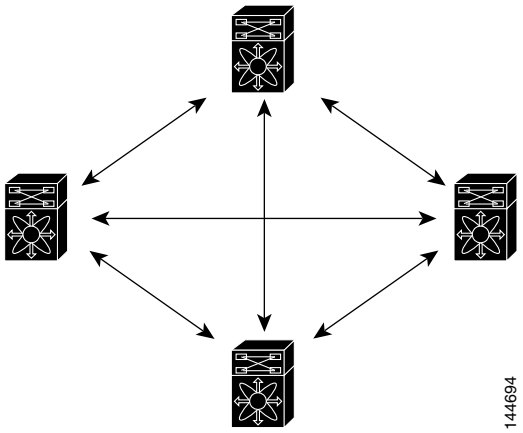
Figure 7-1 Two IPsec Switches Without CAs and Digital Certificates



Every time a new switch is added to the IPsec network, you must configure keys between the new switch and each of the existing switches. (In Figure 7-2, four additional two-part key configurations are required to add a single encrypting switch to the network).

Consequently, the more devices that require IPsec services, the more involved the key administration becomes. This approach does not scale well for larger, more complex encrypting networks.

Figure 7-2 Four IPsec Switches Without a CA and Digital Certificates

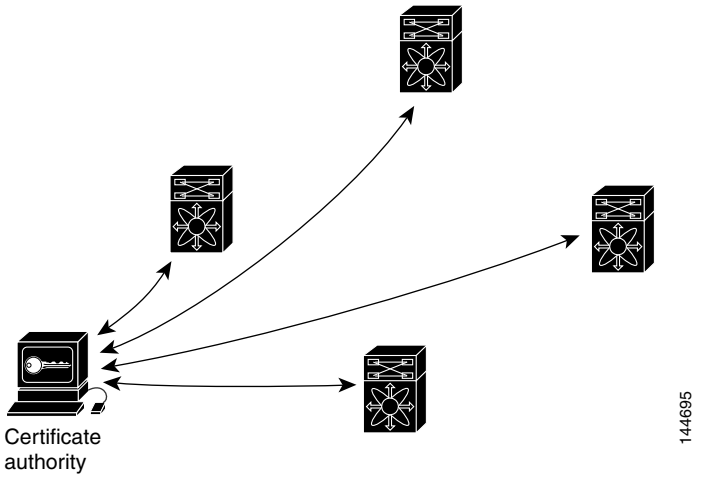


Implementing IPsec with CAs and Digital Certificates

With CA and digital certificates, you do not have to configure keys between all the encrypting switches. Instead, you individually enroll each participating switch with the CA, requesting a certificate for the switch. When this has been accomplished, each participating switch can dynamically authenticate all the other participating switches. When two devices want to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new device is added to the network, you simply enroll that device with a CA, and none of the other devices needs modification. When the new device attempts an IPsec connection, certificates are automatically exchanged and the device can be authenticated.

Figure 7-3 shows the process of dynamically authenticating the devices.

Figure 7-3 Dynamically Authenticating Devices with a CA



To add a new IPsec switch to the network, you need only configure that new switch to request a certificate from the CA, instead of making multiple key configurations with all the other existing IPsec switches.

How CA Certificates Are Used by IPsec Devices

When two IPsec switches want to exchange IPsec-protected traffic passing between them, they must first authenticate each other—otherwise, IPsec protection cannot occur. The authentication is done with IKE.

IKE can use two methods to authenticate the switches, using preshared keys without a CA and using RSA key-pairs with a CA. Both methods require that keys must be preconfigured between the two switches.

Without a CA, a switch authenticates itself to the remote switch using either RSA-encrypted preshared keys.

With a CA, a switch authenticates itself to the remote switch by sending a certificate to the remote switch and performing some public key cryptography. Each switch must send its own unique certificate that was issued and validated by the CA. This process works because the certificate of each switch encapsulates the public key of the switch, each certificate is authenticated by the CA, and all participating switches recognize the CA as an authenticating authority. This scheme is called IKE with an RSA signature.

Your switch can continue sending its own certificate for multiple IPsec sessions, and to multiple IPsec peers until the certificate expires. When the certificate expires, the switch administrator must obtain a new one from the CA.

CAs can also revoke certificates for devices that will no longer participate in IPsec. Revoked certificates are not recognized as valid by other IPsec devices. Revoked certificates are listed in a certificate revocation list (CRL), which each peer may check before accepting a certificate from another peer.

Certificate support for IKE has the following considerations:

- The switch FQDN (host name and domain name) must be configured before installing certificates for IKE.
- Only those certificates that are configured for IKE or general usage are used by IKE.
- The first IKE or general usage certificate configured on the switch is used as the default certificate by IKE.
- The default certificate is for all IKE peers unless the peer specifies another certificate.

- If the peer asks for a certificate which is signed by a CA that it trusts, then IKE uses that certificate, if it exists on the switch, even if it is not the default certificate.
- If the default certificate is deleted, the next IKE or general usage certificate, if any exists, is used by IKE as the default certificate.
- Certificate chaining is not supported by IKE.
- IKE only sends the identity certificate, not the entire CA chain. For the certificate to be verified on the peer, the same CA chain must also exist there.

Manually Configuring IPsec and IKE

This section describes how to manually configure IPsec and IKE.

IPsec provides secure data flows between participating peers. Multiple IPsec data flows can exist between two peers to secure different data flows, with each tunnel using a separate set of SAs.

After you have completed IKE configuration, configure IPsec.

To configure IPsec in each participating IPsec peer, follow these steps:

Step 1 Identify the peers for the traffic to which secure tunnels should be established.

Step 2 Configure the transform set with the required protocols and algorithms.

Step 3 Create the crypto map and apply access control lists (IPv4-ACLs), transform sets, peers, and lifetime values as applicable.

Step 4 Apply the crypto map to the required interface.

This section contains the following topics:

About IKE Initialization

The IKE feature must first be enabled and configured so the IPsec feature can establish data flow with the required peer. Fabric Manager initializes IKE when you first configure it.

You cannot disable IKE if IPsec is enabled. If you disable the IKE feature, the IKE configuration is cleared from the running configuration.

Enabling IKE

To enable IKE, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# feature crypto ike`	Enables the IKE feature.
	`switch(config)# no feature crypto ike`	Disables (default) the IKE feature. **Note** You must disable IPsec before you can disable the IKE feature.

About the IKE Domain

You must apply the IKE configuration to an IPsec domain to allow traffic to reach the supervisor module in the local switch. Fabric Manager sets the IPsec domain automatically when you configure IKE.

Configuring the IKE Domain

You must apply the IKE configurations to an IPsec domain to allow traffic to reach the supervisor module in the local switch.

To configure the IPsec domain, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# crypto ike domain ipsec`	Allows IKE configurations for IPsec domains.

About IKE Tunnels

An IKE tunnel is a secure IKE session between two endpoints. IKE creates this tunnel to protect IKE messages used in IPsec SA negotiations.

Two versions of IKE are used in the Cisco NX-OS implementation.

- IKE version 1 (IKEv1) is implemented using RFC 2407, 2408, 2409, and 2412.
- IKE version 2 (IKEv2) is a simplified and more efficient version and does not interoperate with IKEv1. IKEv2 is implemented using the draft-ietf-ipsec-ikev2-16.txt draft.

About IKE Policy Negotiation

To protect IKE negotiations, each IKE negotiation begins with a common (shared) IKE policy. An IKE policy defines a combination of security parameters to be used during the IKE negotiation. By default, no IKE policy is configured. You must create IKE policies at each peer. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how peers are authenticated. You can create multiple, prioritized policies at each peer to ensure that at least one policy will match a remote peer's policy.

You can configure the policy based on the encryption algorithm (DES, 3DES, or AES), the hash algorithm (SHA or MD5), and the DH group (1, 2, or 5). Each policy can contain a different combination of parameter values. A unique priority number identifies the configured policy. This number ranges from 1 (highest priority) to 255 (lowest priority). You can create multiple policies in a switch. If you need to connect to a remote peer, you must ascertain that at least one policy in the local switch contains the identical parameter values configured in the remote peer. If several policies have identical parameter configurations, the policy with the lowest number is selected.

Table 7-2 provides a list of allowed transform combinations.

Table 7-2 IKE Transform Configuration Parameters

Parameter	Accepted Values	Keyword	Default Value
encryption algorithm	56-bit DES-CBC	**des**	**3des**
	168-bit DES	**3des**	
	128-bit AES	**aes**	
hash algorithm	SHA-1 (HMAC variant)	**sha**	**sha**
	SHA-2 (HMAC variant)	**sha256**	
	MD5 (HMAC variant)	**sha512**	
		md5	
authentication method	Preshared keys	Not configurable	Preshared keys
DH group identifier	768-bit DH	**1**	**1**
	1024-bit DH	**2**	
	1536-bit DH	**5**	

The following table lists the supported and verified settings for IPsec and IKE encryption authentication algorithms on the Microsoft Windows and Linux platforms:

Platform	IKE	IPsec
Microsoft iSCSI initiator, Microsoft IPsec implementation on Microsoft Windows 2000 platform	3DES, SHA-1, SHA-2, or MD5, DH group 2	3DES, SHA-1, SHA-2
Cisco iSCSI initiator, Free Swan IPsec implementation on Linux platform	3DES, MD5, DH group 1	3DES, MD5

Note When you configure the hash algorithm, the corresponding HMAC version is used as the authentication algorithm.

When the IKE negotiation begins, IKE looks for an IKE policy that is the same on both peers. The peer that initiates the negotiation will send all its policies to the remote peer, and the remote peer will try to find a match. The remote peer looks for a match by comparing its own highest priority policy against the other peer's received policies. The remote peer checks each of its policies in order of its priority (highest priority first) until a match is found.

A match is found when the two peers have the same encryption, hash algorithm, authentication algorithm, and DH group values. If a match is found, IKE completes the security negotiation and the IPsec SAs are created.

If an acceptable match is not found, IKE refuses negotiation and the IPsec data flows will not be established.

Configuring an IKE Policy

To configure the IKE negotiation parameters, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# crypto ike domain ipsec`	Allows IPsec domains to be configured in this switch.
Step 3	`switch(config-ike-ipsec)# identity address`	Configures the identity mode for the IKE protocol to use the IP address (default).
	`switch(config-ike-ipsec)# identity hostname`	Configures the identity mode for the IKE protocol to use the fully-qualified domain name (FQDN). **Note** The FQDN is required for using RSA signatures for authentication.
Step 4	`switch(config-ike-ipsec)# key switch1 address 10.10.1.1`	Associates a preshared key with the IP address of a peer.
	`switch(config-ike-ipsec)# key switch1 hostname switch1.cisco.com`	Associates a preshared key with the FQDN of a peer. **Note** To use the FQDN, you must configure the switch name and domain name on the peer.
Step 5	`switch(config-ike-ipsec)# policy 1`	Specifies the policy to configure.
Step 6	`switch(config-ike-ipsec-policy)# encryption des`	Configures the encryption policy.
Step 7	`switch(config-ike-ipsec-policy)# group 5`	Configures the DH group.
Step 8	`switch(config-ike-ipsec-policy)# hash md5`	Configures the hash algorithm.
Step 9	`switch(config-ike-ipsec-policy)# authentication pre-share`	Configures the authentication method to use the preshared key (default).
	`switch(config-ike-ipsec-policy)# authentication rsa-sig`	Configures the authentication method to use the RSA signature. **Note** To use RSA signatures for authentication you must configure identity authentication mode using the FQDN (see Step 3).

Note When the authentication method is rsa-sig, make sure the identity hostname is configured for IKE because the IKE certificate has a subject name of the FQDN type.

Note Before you downgrade to Cisco MDS NX-OS Release 5.2(x), unconfigure the preshared key. Once downgrading is complete, reconfigure the preshared key using the **key** *key-name* **hostname** *host* or **key** *key-name* **address** *ip-address* commands.

Optional IKE Parameter Configuration

You can optionally configure the following parameters for the IKE feature:

- The lifetime association within each policy—The lifetime ranges from 600 to 86,400 seconds. The default is 86,400 seconds (equals one day). The lifetime association within each policy is configured when you are creating an IKE policy. See the "Configuring an IKE Policy" section on page 7-183.
- The keepalive time for each peer if you use IKEv2—The keepalive ranges from 120 to 86,400 seconds. The default is 3,600 seconds (equals one hour).
- The initiator version for each peer—IKE v1 or IKE v2 (default). Your choice of initiator version does not affect interoperability when the remote device initiates the negotiation. Configure this option if the peer device supports IKEv1 and you can play the initiator role for IKE with the specified device. Use the following considerations when configuring the initiator version with FCIP tunnels:
  - If the switches on both sides of an FCIP tunnel are running MDS SAN-OS Release 3.0(1) or later, or Cisco NX-OS 4.1(1) you must configure initiator version IKEv1 on both sides of an FCIP tunnel to use only IKEv1. If one side of an FCIP tunnel is using IKEv1 and the other side is using IKEv2, the FCIP tunnel uses IKEv2.
  - If the switch on one side of an FCIP tunnel is running MDS SAN-OS Release 3.0(1) or later, or Cisco NX-OS 4.1(1b) and the switch on the other side of the FCIP tunnel is running MDS SAN-OS Release 2.x, configuring IKEv1 on either side (or both) results in the FCIP tunnel using IKEv1.

Note Only IKE v1 is supported to build IPsec between 2.x and 3.x MDS switches.

Caution You may need to configure the initiator version even when the switch does not behave as an IKE initiator under normal circumstances. Always using this option guarantees a faster recovery of traffic flows in case of failures.

Tip The keepalive time only applies to IKEv2 peers and not to all peers.

Note When IPsec implementations in the host prefer to initiate the IPsec rekey, be sure to configure the IPsec lifetime value in the Cisco MDS switch to be higher than the lifetime value in the host.

This section includes the following topics:

Configuring the Lifetime Association for a Policy

To configure the lifetime association for each policy, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# crypto ike domain ipsec`	Allows IPsec domains to be configured in this switch.
Step 3	`switch(config-ike-ipsec)# policy 1`	Specifies the policy to configure.
Step 4	`switch(config-ike-ipsec-policy) lifetime seconds 6000`	Configures a lifetime of 6,000 seconds.

Configuring the Keepalive Time for a Peer

To configure the keepalive time for each peer, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# crypto ike domain ipsec`	Allows IPsec domains to be configured in this switch.
Step 3	`switch(config-ike-ipsec)# keepalive 60000`	Configures the keepalive time for all peers to be 60,000 seconds.

Configuring the Initiator Version

To configure the initiator version using IPv4, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# crypto ike domain ipsec`	Allows IPsec domains to be configured in this switch.
Step 3	`switch(config-ike-ipsec)# initiator version 1 address 10.10.10.1`	Configures the switch to use IKEv1 when initiating IKE with device 10.10.10.0 **Note** IKE supports IPv4 addresses, not IPv6 addresses.

Clearing IKE Tunnels or Domains

If an IKE tunnel ID is not specified for the IKE configuration, you can clear all existing IKE domain connections by issuing the **clear crypto ike domain ipsec sa** command in EXEC mode.

```
switch# clear crypto ike domain ipsec sa
```

Caution When you delete all the SAs within a specific IKEv2 tunnel, then that IKE tunnel is automatically deleted.

If an SA is specified for the IKE configuration, you can clear the specified IKE tunnel ID connection by issuing the **clear crypto ike domain ipsec sa** *IKE_tunnel-ID* command in EXEC mode.

```
switch# clear crypto ike domain ipsec sa 51
```

Caution When you delete the IKEv2 tunnel, the associated IPsec tunnel under that IKE tunnel is automatically deleted.

Refreshing SAs

Use the **crypto ike domain ipsec rekey IPv4-ACL**-*index* command to refresh the SAs after performing IKEv2 configuration changes.

Crypto IPv4-ACLs

IP access control lists (IPv4-ACLs) provide basic network security to all switches in the Cisco MDS 9000 Family. IPv4 IP-ACLs restrict IP-related traffic based on the configured IP filters. See Chapter 5, "Configuring IPv4 and IPv6 Access Control Lists" for details on creating and defining IPv4-ACLs.

In the context of crypto maps, IPv4-ACLs are different from regular IPv4-ACLs. Regular IPv4-ACLs determine what traffic to forward or block at an interface. For example, IPv4-ACLs can be created to protect all IP traffic between subnet A and subnet Y or Telnet traffic between host A and host B.

This section contains the following topics:

About Crypto IPv4-ACLs

Crypto IPv4-ACLs are used to define which IP traffic requires crypto protection and which traffic does not.

Crypto IPv4-ACLs associated with IPsec crypto map entries have four primary functions:

- Select outbound traffic to be protected by IPsec (permit = protect).
- Indicate the data flow to be protected by the new SAs (specified by a single permit entry) when initiating negotiations for IPsec SAs.
- Process inbound traffic to filter out and discard traffic that should have been protected by IPsec.
- Determine whether or not to accept requests for IPsec SAs on behalf of the requested data flows when processing IKE negotiation from the IPsec peer.

Tip If you want some traffic to receive one type of IPsec protection (for example, encryption only) and other traffic to receive a different type of IPsec protection (for example, both authentication and encryption), create two IPv4-ACLs. Use both IPv4-ACLs in different crypto maps to specify different IPsec policies.

Note IPsec does not support IPv6-ACLs.

Crypto IPv4-ACL Guidelines

Follow these guidelines when configuring IPv4-ACLs for the IPsec feature:

- The Cisco NX-OS software only allows name-based IPv4-ACLs.
- When an IPv4-ACL is applied to a crypto map, the following options apply:
  - Permit—Applies the IPsec feature to the traffic.
  - Deny—Allows clear text (default).

  Note IKE traffic (UDP port 500) is implicitly transmitted in clear text.

- The IPsec feature only considers the source and destination IPv4 addresses and subnet masks, protocol, and single port number. There is no support for IPv6 in IPsec.

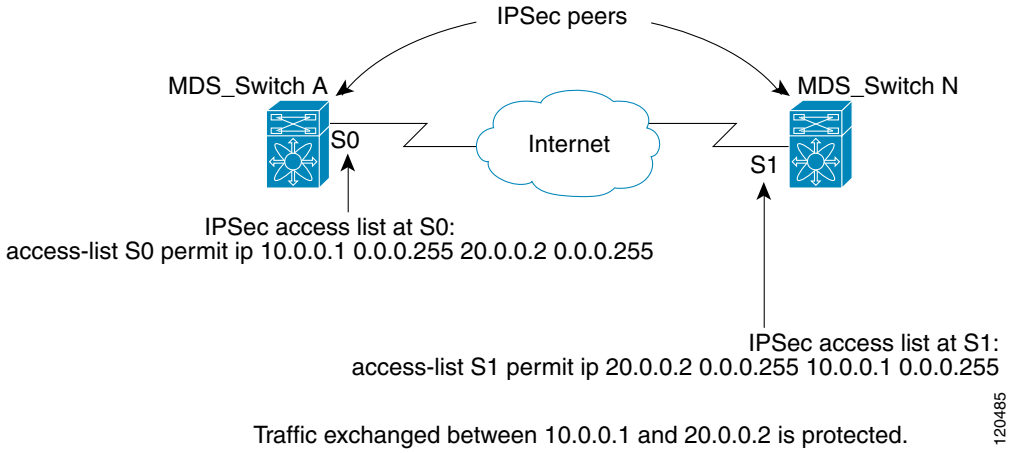
  Note The IPsec feature does not support port number ranges and ignores higher port number field, if specified.

- The permit option causes all IP traffic that matches the specified conditions to be protected by crypto, using the policy described by the corresponding crypto map entry.
- The deny option prevents traffic from being protected by crypto. The first deny statement causes the traffic to be in clear text.
- The crypto IPv4-ACL you define is applied to an interface after you define the corresponding crypto map entry and apply the crypto map set to the interface.
- Different IPv4-ACLs must be used in different entries of the same crypto map set.

- Inbound and outbound traffic is evaluated against the same outbound IPv4-ACL. Therefore, the IPv4-ACL's criteria is applied in the forward direction to traffic exiting your switch, and the reverse direction to traffic entering your switch.
- Each IPv4-ACL filter assigned to the crypto map entry is equivalent to one security policy entry.
- IPsec protection (see Figure 7-4) is applied to traffic between switch interface S0 (IPv4 address 10.0.0.1) and switch interface S1 (IPv4 address 20.0.0.2) as the data exits switch A's S0 interface enroute to switch interface S1. For traffic from 10.0.0.1 to 20.0.0.2, the IPv4-ACL entry on switch A is evaluated as follows:
  - source = IPv4 address 10.0.0.1
  - dest = IPv4 address 20.0.0.2

  For traffic from 20.0.0.2 to 10.0.0.1, that same IPv4-ACL entry on switch A is evaluated as follows:
  - source = IPv4 address 20.0.0.2
  - dest = IPv4 address 10.0.0.1

Figure 7-4 IPsec Processing of Crypto IPv4-ACLs



- If you configure multiple statements for a given crypto IPv4-ACL that is used for IPsec, the first permit statement that is matched is used to determine the scope of the IPsec SA. Later, if traffic matches a different permit statement of the crypto IPv4-ACL, a new, separate IPsec SA is negotiated to protect traffic matching the newly matched IPv4-ACL statement.
- Unprotected inbound traffic that matches a permit entry in the crypto IPv4-ACL for a crypto map entry flagged as IPsec is dropped, because this traffic was expected to be protected by IPsec.
- You can use the **show ip access-lists** command to view all IP-ACLs. The IP-ACLs used for traffic filtering purposes are also used for crypto.
- For IPsec to interoperate effectively with Microsoft iSCSI initiators, specify the TCP protocol and the local iSCSI TCP port number (default 3260) in the IPv4-ACL. This configuration ensures the speedy recovery of encrypted iSCSI sessions following disruptions such as Gigabit Ethernet interfaces shutdowns, VRRP switchovers, and port failures.
- The following example of a IPv4-ACL entry shows that the MDS switch IPv4 address is 10.10.10.50 and remote Microsoft host running encrypted iSCSI sessions is 10.10.10.16:

```
switch(config)# ip access-list aclmsiscsi2 permit tcp 10.10.10.50 0.0.0.0 range port
3260 3260 10.10.10.16 0.0.0.0
```

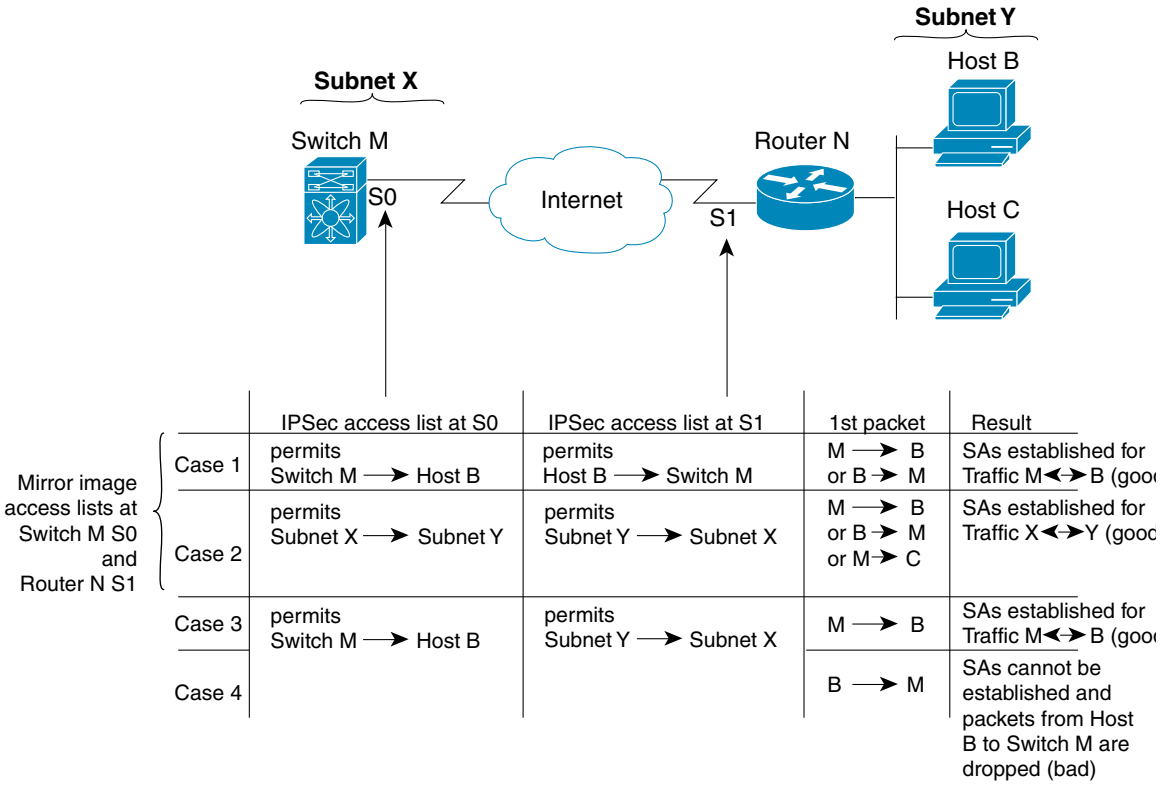
Mirror Image Crypto IPv4-ACLs

For every crypto IPv4-ACL specified for a crypto map entry defined at the local peer, define a mirror image crypto IPv4-ACL at the remote peer. This configuration ensures that IPsec traffic applied locally can be processed correctly at the remote peer.

Tip The crypto map entries themselves must also support common transforms and must refer to the other system as a peer.

Figure 7-5 shows some sample scenarios with and without mirror image IPv4-ACLs.

Figure 7-5 IPsec Processing of Mirror Image Configuration



As Figure 7-5 indicates, IPsec SAs can be established as expected whenever the two peers' crypto IPv4-ACLs are mirror images of each other. However, an IPsec SA can be established only some of the time when the IPv4-ACLs are not mirror images of each other. This can happen in the case when an entry in one peer's IPv4-ACL is a subset of an entry in the other peer's IPv4-ACL, such as shown in cases 3 and 4 of Figure 7-5. IPsec SA establishment is critical to IPsec. Without SAs, IPsec does not work, causing any packets matching the crypto IPv4-ACL criteria to be silently dropped instead of being forwarded with IPsec security.

In case 4, an SA cannot be established because SAs are always requested according to the crypto IPv4-ACLs at the initiating packet's end. In case 4, router N requests that all traffic between subnet X and subnet Y be protected, but this is a superset of the specific flows permitted by the crypto IPv4-ACL at switch M so the request is not permitted. Case 3 works because switch M's request is a subset of the specific flows permitted by the crypto IPv4-ACL at router N.

Because of the complexities introduced when crypto IPv4-ACLs are not configured as mirror images at peer IPsec devices, we strongly encourage you to use mirror image crypto IPv4-ACLs.

The any Keyword in Crypto IPv4-ACLs

Tip We recommend that you configure mirror image crypto IPv4-ACLs for use by IPsec and that you avoid using the **any** option.

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

The **permit any** statement causes all outbound traffic to be protected (and all protected traffic sent to the peer specified in the corresponding crypto map entry) and requires protection for all inbound traffic. Then, all inbound packets that lack IPsec protection are silently dropped, including packets for routing protocols, NTP, echo, echo response, and so forth.

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

Creating Crypto IPv4-ACLs

To create IPv4-ACLs, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# ip access-list List1 permit ip 10.1.1.100 0.0.0.255 11.1.1.100 0.0.0.255`	Permits all IP traffic from and to the specified networks.

Note The **show ip access-list** command does not display the crypto map entries. Use the **show crypto map** command to display the associated entries.

Add permit and deny statements as appropriate (see Chapter 5, “Configuring IPv4 and IPv6 Access Control Lists”). Each permit and deny specifies conditions to determine which IP packets must be protected.

About Transform Sets in IPsec

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

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

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

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

Note When you enable IPsec, the Cisco NX-OS software automatically creates a default transform set (ipsec_default_tranform_set) using AES-128 encryption and SHA-1 authentication algorithms.

Table 7-3 provides a list of allowed transform combinations for IPsec.

Table 7-3 IPsec Transform Configuration Parameters

Parameter	Accepted Values	Keyword
encryption algorithm	56-bit DES-CBC	**esp-des**
	168-bit DES	**esp-3des**
	128-bit AES-CBC	**esp-aes 128**
	128-bit AES-CTR[1]	**esp-aes 128 ctr**
	256-bit AES-CBC	**esp-aes 256**
	256-bit AES-CTR[1]	**esp-aes 256 ctr**
hash/authentication algorithm[1] (optional)	SHA-1 (HMAC variant)	**esp-sha1-hmac**
	SHA-2 (HMAC variant)	**esp-sha256-hmac**
	MD5 (HMAC variant)	**esp-sha512-hmac**
	AES-XCBC-MAC	**esp-md5-hmac**
		esp-aes-xcbc-mac[2]

1. If you configure the AES counter (CTR) mode, you must also configure the authentication algorithm.
2. Starting from Cisco MDS NX-OS Release 5.2(2), the **esp-aes-xcbc-mac** authentication algorithm is not supported.

The following table lists the supported and verified settings for IPsec and IKE encryption authentication algorithms on the Microsoft Windows and Linux platforms:

Platform	IKE	IPsec
Microsoft iSCSI initiator, Microsoft IPsec implementation on Microsoft Windows 2000 platform	3DES, SHA-1, SHA-2, or MD5, DH group 2	3DES, SHA-1, SHA-2
Cisco iSCSI initiator, Free Swan IPsec implementation on Linux platform	3DES, MD5, DH group 1	3DES, MD5

Configuring Transform Sets

To configure transform sets, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# crypto transform-set domain ipsec test esp-3des esp-md5-hmac`	Configures a transform set called test specifying the 3DES encryption algorithm and the MD5 authentication algorithm. Refer to Table 7-3 to verify the allowed transform combinations.
	`switch(config)# crypto transform-set domain ipsec test esp-3des`	Configures a transform set called test specifying the 3DES encryption algorithm. In this case, the default no authentication is performed.

About Crypto Map Entries

Once you have created the crypto IPv4-ACLs and transform sets, you can create crypto map entries that combine the various parts of the IPsec SA, including the following:

- The traffic to be protected by IPsec (per the crypto IPv4-ACL). A crypto map set can contain multiple entries, each with a different IPv4-ACL.
- The granularity of the flow to be protected by a set of SAs.
- The IPsec-protected traffic destination (who the remote IPsec peer is).
- The local address to be used for the IPsec traffic (applying to an interface).
- The IPsec security to be applied to this traffic (selecting from a list of one or more transform sets).
- Other parameters to define an IPsec SA.

Crypto map entries with the same crypto map name (but different map sequence numbers) are grouped into a crypto map set.

When you apply a crypto map set to an interface, the following events occur:

- A security policy database (SPD) is created for that interface.
- All IP traffic passing through the interface is evaluated against the SPD.

If a crypto map entry sees outbound IP traffic that requires protection, an SA is negotiated with the remote peer according to the parameters included in the crypto map entry.

The policy derived from the crypto map entries is used during the negotiation of SAs. If the local switch initiates the negotiation, it will use the policy specified in the crypto map entries to create the offer to be sent to the specified IPsec peer. If the IPsec peer initiates the negotiation, the local switch checks the policy from the crypto map entries and decides whether to accept or reject the peer's request (offer).

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

SA Establishment Between Peers

When two peers try to establish an SA, they must each have at least one crypto map entry that is compatible with one of the other peer's crypto map entries.

For two crypto map entries to be compatible, they must at least meet the following criteria:

- The crypto map entries must contain compatible crypto IPv4-ACLs (for example, mirror image IPv4-ACLs). If the responding peer entry is in the local crypto, the IPv4-ACL must be permitted by the peer's crypto IPv4-ACL.
- The crypto map entries must each identify the other peer or must have auto peer configured.
- If you create more than one crypto map entry for a given interface, use the seq-num of each map entry to rank the map entries: the lower the seq-num, the higher the priority. At the interface that has the crypto map set, traffic is evaluated against higher priority map entries first.
- The crypto map entries must have at least one transform set in common, where IKE negotiations are carried out and SAs are established. During the IPsec SA negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

When a packet matches a permit entry in a particular IPv4-ACL, the corresponding crypto map entry is tagged, and the connections are established.

Crypto Map Configuration Guidelines

When configuring crypto map entries, follow these guidelines:

- The sequence number for each crypto map decides the order in which the policies are applied. A lower sequence number is assigned a higher priority.
- Only one IPv4-ACL is allowed for each crypto map entry (the IPv4-ACL itself can have multiple permit or deny entries).
- When the tunnel endpoint is the same as the destination address, you can use the auto-peer option to dynamically configure the peer.
- For IPsec to interoperate effectively with Microsoft iSCSI initiators, specify the TCP protocol and the local iSCSI TCP port number (default 3260) in the IPv4-ACL. This configuration ensures the speedy recovery of encrypted iSCSI sessions following disruptions such as Gigabit Ethernet interfaces shutdowns, VRRP switchovers, and port failures.

Creating Crypto Map Entries

Note If the peer IP address specified in the crypto map entry is a VRRP IP address on a remote Cisco MDS switch, ensure that the IP address is created using the **secondary** option (see the *Cisco MDS 9000 Family NX-OS IP Services Configuration Guide* for more information).

To create mandatory crypto map entries, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# crypto map  domain ipsec SampleMap 31`	Places you in the crypto map configuration mode for the entry named SampleMap with 31 as its sequence number.
Step 3	`switch(config-crypto-map-ip)# match address SampleAcl`	Names an ACL to determine which traffic should be protected and not protected by IPsec in the context of this crypto map entry.
Step 4	`switch(config-crypto-map-ip)# set peer 10.1.1.1`	Configures a specific peer IPv4 address. **Note** IKE only supports IPv4 addresses, not IPv6 addresses.
Step 5	`switch(config-crypto-map-ip)# set transform-set SampleTransform1 SampleTransmfor2`	Specifies which transform sets are allowed for the specified crypto map entry or entries. List multiple transform sets in order of priority (highest priority first).

About SA Lifetime Negotiation

You can override the global lifetime values (size and time) by configuring an SA-specific lifetime value.

To specify SA lifetime negotiation values, you can optionally configure the lifetime value for a specified crypto map. If you do, this value overrides the globally set values. If you do not specify the crypto map specific lifetime, the global value (or global default) is used.

See the "Global Lifetime Values" section on page 7-197 for more information on global lifetime values.

Setting the SA Lifetime

To set the SA lifetime for a specified crypto map entry, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# crypto map domain ipsec SampleMap 31`	Enters crypto map configuration submode for the entry named SampleMap with 31 as its sequence number.
Step 3	`switch(config-crypto-map-ip)# set security-association lifetime seconds 8640`	Specifies an SA lifetime for this crypto map entry using different IPsec SA lifetimes than the global lifetimes for the crypto map entry.
Step 4	`switch(config-crypto-map-ip)# set security-association lifetime gigabytes 4000`	Configures the traffic-volume lifetime for this SA to time out after the specified amount of traffic (in gigabytes) have passed through the FCIP link using the SA. The lifetime ranges from 1 to 4095 gigabytes.

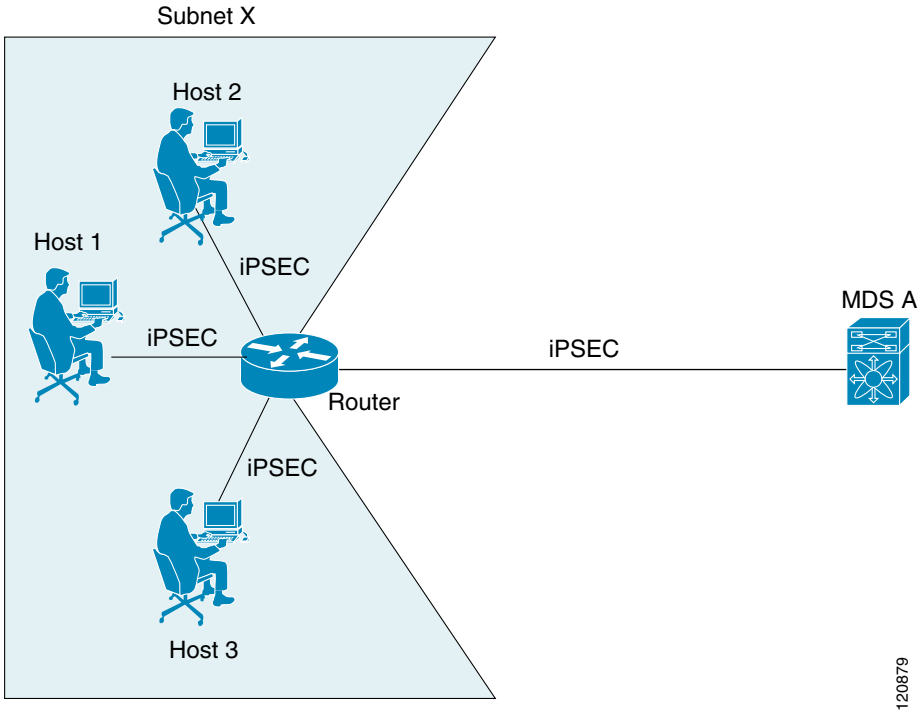
About the AutoPeer Option

Setting the peer address as **auto-peer** in the crypto map indicates that the destination endpoint of the traffic should be used as the peer address for the SA. Using the same crypto map, a unique SA can be set up at each of the endpoints in the subnet specified by the crypto map's IPv4-ACL entry. Auto-peer simplifies configuration when traffic endpoints are IPsec capable. It is particularly useful for iSCSI, where the iSCSI hosts in the same subnet do not require separate configuration.

Figure 7-6 shows a scenario where the auto-peer option can simplify configuration. Using the auto-peer option, only one crypto map entry is needed for all the hosts from subnet X to set up SAs with the switch. Each host will set up its own SA, but will share the crypto map entry. Without the auto-peer option, each host needs one crypto map entry.

See the “Sample iSCSI Configuration” section on page 7-207 for more details.

Figure 7-6 iSCSI with End-to-End IPsec Using the auto-peer Option



Configuring the AutoPeer Option

To configure the auto-peer option, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.

	Command	Purpose
Step 2	`switch(config)# crypto map domain ipsec SampleMap 31`	Places you in the crypto map configuration mode for the entry named SampleMap with 31 as its sequence number.
Step 3	`switch(config-crypto-map-ip)# set peer auto-peer`	Directs the software to select (during the SA setup) the destination peer IP address dynamically.

About Perfect Forward Secrecy

To specify SA lifetime negotiation values, you can also optionally configure the perfect forward secrecy (PFS) value in the crypto map.

The PFS feature is disabled by default. If you set the PFS group, you can set one of the DH groups: 1, 2, 5, or 14. If you do not specify a DH group, the software uses group 1 by default.

Configuring Perfect Forward Secrecy

To configure the PFS value, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# crypto map domain ipsec SampleMap 31`	Places you in the crypto map configuration mode for the entry named SampleMap with 31 as its sequence number.
Step 3	`switch(config-crypto-map-ip)# set pfs group 2`	Specifies that IPsec should ask for PFS when requesting new SAs for this crypto map entry, or should demand PFS in requests received from the IPsec peer.

About Crypto Map Set Interface Application

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

You can apply only one crypto map set to an interface. You can apply the same crypto map to multiple interfaces. However, you cannot apply more than one crypto map set to each interface.

Applying a Crypto Map Set

To apply a crypto map set to an interface, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# interface gigabitethernet 4/1`	Selects the required Gigabit Ethernet interface (and subinterface, if required) to which the IPsec crypto map is to be applied.
Step 3	`switch(config-if)# crypto map domain ipsec cm10`	Applies the crypto map set to the selected interface.

IPsec Maintenance

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

Tip You can obtain the SA index from the output of the **show crypto sa domain interface gigabitethernet** *slot/port* command.

Use the following command to clear part of the SA database.

```
switch# clear crypto sa domain ipsec interface gigabitethernet 2/1 inbound sa-index 1
```

Global Lifetime Values

If you have not configured a lifetime in the crypto map entry, the global lifetime values are used when negotiating new IPsec SAs.

You can configure two lifetimes: timed or traffic-volume. An SA expires after the first of these lifetimes is reached. The default lifetimes are 3,600 seconds (one hour) and 450 GB.

If you change a global lifetime, the new lifetime value will not be applied to currently existing SAs, but will be used in the negotiation of subsequently established SAs. If you wish to use the new values immediately, you can clear all or part of the SA database.

Assuming that the particular crypto map entry does not have lifetime values configured, when the switch requests new SAs it will specify its global lifetime values in the request to the peer; it will use this value as the lifetime of the new SAs. When the switch receives a negotiation request from the peer, it uses the value determined by the IKE version in use:

- If you use IKEv1 to set up IPsec SAs, the SA lifetime values are chosen to be the smaller of the two proposals. The same values are programmed on both the ends of the tunnel.

- If you use IKEv2 to set up IPsec SAs, the SAs on each end have their own set up of lifetime values and thus the SAs on both sides expire independently.

The SA (and corresponding keys) will expire according to whichever comes sooner, either after the specified amount of time (in seconds) has passed or after the specified amount of traffic (in bytes) has passed.

A new SA is negotiated before the lifetime threshold of the existing SA is reached to ensure that negotiation completes before the existing SA expires.

The new SA is negotiated when one of the following thresholds is reached (whichever comes first):

- 30 seconds before the lifetime expires or
- Approximately 10% of the lifetime in bytes remain

If no traffic has passed through when the lifetime expires, a new SA is not negotiated. Instead, a new SA will be negotiated only when IPsec sees another packet that should be protected.

To configure global SA lifetimes, follow these steps:

	Command	Purpose
Step 1	`switch# config terminal`	Enters configuration mode.
Step 2	`switch(config)# crypto global domain ipsec security-association lifetime seconds 86400`	Configures the global timed lifetime for IPsec SAs to time out after the specified number of seconds have passed. The global lifetime ranges from 120 to 86400 seconds.
Step 3	`switch(config)# crypto global domain ipsec security-association lifetime gigabytes 4000`	Configures the global traffic-volume lifetime for IPsec SAs to time out after the specified amount of traffic (in gigabytes) has passed through the FCIP link using the SA. The global lifetime ranges from 1 to 4095 gigabytes.
	`switch(config)# crypto global domain ipsec security-association lifetime kilobytes 2560`	Configures the global traffic-volume lifetime in kilobytes. The global lifetime ranges from 2560 to 2147483647 kilobytes.
	`switch(config)# crypto global domain ipsec security-association lifetime megabytes 5000`	Configures the global traffic-volume lifetime in megabytes. The global lifetime ranges from 3 to 4193280 megabytes.

Displaying IKE Configurations

You can verify the IKE information by using the **show** set of commands. See Examples 7-1 to 7-5.

Example 7-1 Displays the Parameters Configured for Each IKE Policy

```
switch# show crypto ike domain ipsec
keepalive 60000
```

Example 7-2 Displays the Initiator Configuration

```
switch# show crypto ike domain ipsec initiator
initiator version 1 address 1.1.1.1
initiator version 1 address 1.1.1.2
```

Example 7-3 Displays the Key Configuration

```
switch# show crypto ike domain ipsec key
key abcdefgh address 1.1.1.1
key bcdefghi address 1.1.2.1
```

Example 7-4 Displays the Currently Established Policies for IKE

```
switch# show crypto ike domain ipsec policy 1
Priority 1, auth pre-shared, lifetime 6000 secs, encryption 3des, hash md5, DH group 5
Priority 3, auth pre-shared, lifetime 86300 secs, encryption aes, hash sha1, DH group 1
Priority 5, auth pre-shared-key, lifetime 86400 secs, encryption 3des, hash sha256, DH
group 1
```

Example 7-5 Displays the Currently Established SAs for IKE

```
switch# show crypto ike domain ipsec sa
Tunn    Local Addr          Remote Addr        Encr     Hash    Auth Method     Lifetime
------------------------------------------------------------------------------------------
1*      172.22.31.165[500]  172.22.31.166[500] 3des     sha1    preshared key   86400
2       172.22.91.174[500]  172.22.91.173[500] 3des     sha1    preshared key   86400
------------------------------------------------------------------------------------------
NOTE: tunnel id ended with * indicates an IKEv1 tunnel
```

Displaying IPsec Configurations

You can verify the IPsec information by using the **show** set of commands. See Examples 7-6 to 7-19.

Example 7-6 Displays Information for the Specified ACL

```
switch# show ip access-list acl10
ip access-list acl10 permit ip 10.10.10.0 0.0.0.255 10.10.10.0 0.0.0.255 (0 matches)
```

In Example 7-6, the display output match is only displayed of an interface (not the crypto map) meets this criteria.

Example 7-7 Displays the Transform Set Configuration

```
switch# show crypto transform-set domain ipsec
Transform set: 1/1 {esp-3des esp-sha256-hmac}
    will negotiate {tunnel}
Transform set: ipsec_default_transform_set {esp-aes 128 esp-sha1-hmac}
    will negotiate {tunnel}
```

Example 7-8 Displays All Configured Crypto Maps

```
switch# show crypto map domain ipsec
Crypto Map "cm10" 1 ipsec
        Peer = Auto Peer
        IP ACL = acl10
            permit ip 10.10.10.0 255.255.255.0 10.10.10.0 255.255.255.0
        Transform-sets: 3des-md5, des-md5,
        Security Association Lifetime: 4500 megabytes/3600 seconds
        PFS (Y/N): N
        Interface using crypto map set cm10:
```

```
            GigabitEthernet4/1
Crypto Map "cm100" 1 ipsec
        Peer = Auto Peer
        IP ACL = acl100
            permit ip 10.10.100.0 255.255.255.0 10.10.100.0 255.255.255.0
        Transform-sets: 3des-md5, des-md5,
        Security Association Lifetime: 4500 megabytes/3600 seconds
        PFS (Y/N): N
        Interface using crypto map set cm100:
            GigabitEthernet4/2
```

Example 7-9 Displays the Crypto Map Information for a Specific Interface

```
switch# show crypto map domain ipsec interface gigabitethernet 4/1
Crypto Map "cm10" 1 ipsec
        Peer = Auto Peer
        IP ACL = acl10
            permit ip 10.10.10.0 255.255.255.0 10.10.10.0 255.255.255.0
        Transform-sets: 3des-md5, des-md5,
        Security Association Lifetime: 4500 megabytes/3600 seconds
        PFS (Y/N): N
        Interface using crypto map set cm10:
            GigabitEthernet4/1
```

Example 7-10 Displays the Specified Crypto Map Information

```
switch# show crypto map domain ipsec tag cm100
Crypto Map "cm100" 1 ipsec
        Peer = Auto Peer
        IP ACL = acl100
            permit ip 10.10.100.0 255.255.255.0 10.10.100.0 255.255.255.0
        Transform-sets: 3des-md5, des-md5,
        Security Association Lifetime: 4500 megabytes/3600 seconds
        PFS (Y/N): N
        Interface using crypto map set cm100:
            GigabitEthernet4/2
```

Example 7-11 Displays SA Association for the Specified Interface

```
switch# show crypto sad domain ipsec interface gigabitethernet 4/1
interface: GigabitEthernet4/1
    Crypto map tag: cm10, local addr. 10.10.10.1
    protected network:
    local ident (addr/mask): (10.10.10.0/255.255.255.0)
    remote  ident (addr/mask): (10.10.10.4/255.255.255.255)
    current_peer: 10.10.10.4
      local crypto endpt.: 10.10.10.1, remote crypto endpt.: 10.10.10.4
      mode: tunnel, crypto algo: esp-3des, auth algo: esp-md5-hmac
     current outbound spi: 0x30e000f (51249167), index: 0
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 423624704
     current inbound spi: 0x30e0000 (51249152), index: 0
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 423624704
```

Example 7-12 Displays All SA Associations

```
switch# show crypto sad domain ipsec
interface: GigabitEthernet4/1
    Crypto map tag: cm10, local addr. 10.10.10.1
    protected network:
    local  ident (addr/mask): (10.10.10.0/255.255.255.0)
    remote  ident (addr/mask): (10.10.10.4/255.255.255.255)
    current_peer: 10.10.10.4
      local crypto endpt.: 10.10.10.1, remote crypto endpt.: 10.10.10.4
      mode: tunnel, crypto algo: esp-3des, auth algo: esp-md5-hmac
     current outbound spi: 0x30e000f (51249167), index: 0
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 423624704
     current inbound spi: 0x30e0000 (51249152), index: 0
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 423624704
```

Example 7-13 Displays Information About the Policy Database

```
switch# show crypto spd domain ipsec
Policy Database for interface: GigabitEthernet4/1, direction: Both
#   0:      deny  udp any port eq 500 any
#   1:      deny  udp any any port eq 500
#   2:      permit ip 10.10.10.0 255.255.255.0 10.10.10.0 255.255.255.0
#  63:      deny  ip any any
Policy Database for interface: GigabitEthernet4/2, direction: Both
#   0:      deny  udp any port eq 500 any <----------------------UDP default entry
#   1:      deny  udp any any port eq 500 <---------------------- UDP default entry
#   3:      permit ip 10.10.100.0 255.255.255.0 10.10.100.0 255.255.255.0
#  63:      deny  ip any any <---------------------------------------- Clear text default entry
```

Example 7-14 Displays SPD Information for a Specific Interface

```
switch# show crypto spd domain ipsec interface gigabitethernet 4/2
Policy Database for interface: GigabitEthernet3/1, direction: Both
#   0:      deny  udp any port eq 500 any
#   1:      deny  udp any any port eq 500
#   2:      permit ip 10.10.10.0 255.255.255.0 10.10.10.0 255.255.255.0
# 127:      deny  ip any any
```

Example 7-15 Displays Detailed iSCSI Session Information for a Specific Interface

```
switch# show iscsi session detail
Initiator iqn.1987-05.com.cisco:01.9f39f09c7468 (ips-host16.cisco.com)
    Initiator ip addr (s): 10.10.10.5
  Session #1 (index 24)
    Discovery session, ISID 00023d000001, Status active

  Session #2 (index 25)
    Target ibm1
    VSAN 1, ISID 00023d000001, TSIH 0, Status active, no reservation
    Type Normal, ExpCmdSN 42, MaxCmdSN 57, Barrier 0
    MaxBurstSize 0, MaxConn 1, DataPDUInOrder Yes
    DataSeqInOrder Yes, InitialR2T Yes, ImmediateData No
    Registered LUN 0, Mapped LUN 0
    Stats:
      PDU: Command: 41, Response: 41
      Bytes: TX: 21388, RX: 0
```

```
    Number of connection: 1
    Connection #1
      iSCSI session is protected by IPSec <----------The iSCSI session protection status
      Local IP address: 10.10.10.4, Peer IP address: 10.10.10.5
      CID 0, State: Full-Feature
      StatSN 43, ExpStatSN 0
      MaxRecvDSLength 131072, our_MaxRecvDSLength 262144
      CSG 3, NSG 3, min_pdu_size 48 (w/ data 48)
      AuthMethod none, HeaderDigest None (len 0), DataDigest None (len 0)
      Version Min: 0, Max: 0
      FC target: Up, Reorder PDU: No, Marker send: No (int 0)
      Received MaxRecvDSLen key: Yes
```

Example 7-16 Displays FCIP Information for a Specific Interface

```
switch# show interface fcip 1
fcip1 is trunking
    Hardware is GigabitEthernet
    Port WWN is 20:50:00:0d:ec:08:6c:c0
    Peer port WWN is 20:10:00:05:30:00:a7:9e
    Admin port mode is auto, trunk mode is on
    Port mode is TE
    Port vsan is 1
    Speed is 1 Gbps
    Trunk vsans (admin allowed and active) (1)
    Trunk vsans (up)                       (1)
    Trunk vsans (isolated)                 ()
    Trunk vsans (initializing)             ()
    Using Profile id 1  (interface GigabitEthernet2/1)
    Peer Information
      Peer Internet address is 10.10.11.1 and port is 3225
    FCIP tunnel is protected by IPSec <----------The FCIP tunnel protection status
    Write acceleration mode is off
    Tape acceleration mode is off
    Tape Accelerator flow control buffer size is 256 KBytes
    IP Compression is disabled
    Special Frame is disabled
    Maximum number of TCP connections is 2
    Time Stamp is disabled
    QOS control code point is 0
    QOS data code point is 0
    B-port mode disabled
    TCP Connection Information
      2 Active TCP connections
        Control connection: Local 10.10.11.2:3225, Remote 10.10.11.1:65520
        Data connection: Local 10.10.11.2:3225, Remote 10.10.11.1:65522
      2 Attempts for active connections, 0 close of connections
    TCP Parameters
      Path MTU 1400 bytes
      Current retransmission timeout is 200 ms
      Round trip time: Smoothed 2 ms, Variance: 1
      Advertized window: Current: 124 KB, Maximum: 124 KB, Scale: 6
      Peer receive window: Current: 123 KB, Maximum: 123 KB, Scale: 6
      Congestion window: Current: 53 KB, Slow start threshold: 48 KB
      Current Send Buffer Size: 124 KB, Requested Send Buffer Size: 0 KB
      CWM Burst Size: 50 KB
    5 minutes input rate 128138888 bits/sec, 16017361 bytes/sec, 7937 frames/sec
    5 minutes output rate 179275536 bits/sec, 22409442 bytes/sec, 46481 frames/sec
      10457037 frames input, 21095415496 bytes
         308 Class F frames input, 32920 bytes
         10456729 Class 2/3 frames input, 21095382576 bytes
         9907495 Reass frames
```

```
   0 Error frames timestamp error 0
63792101 frames output, 30250403864 bytes
   472 Class F frames output, 46816 bytes
   63791629 Class 2/3 frames output, 30250357048 bytes
   0 Error frames
```

Example 7-17 Displays the Global IPsec Statistics for the Switch

```
switch# show crypto global domain ipsec
IPSec global statistics:
        Number of crypto map sets: 3
        IKE transaction stats: 0 num, 256 max
        Inbound SA stats: 0 num
        Outbound SA stats: 0 num
```

Example 7-18 Displays the IPsec Statistics for the Specified Interface

```
switch# show crypto global domain ipsec interface gigabitethernet 3/1
IPSec interface statistics:
        IKE transaction stats: 0 num
        Inbound SA stats: 0 num, 512 max
        Outbound SA stats: 0 num, 512 max
```

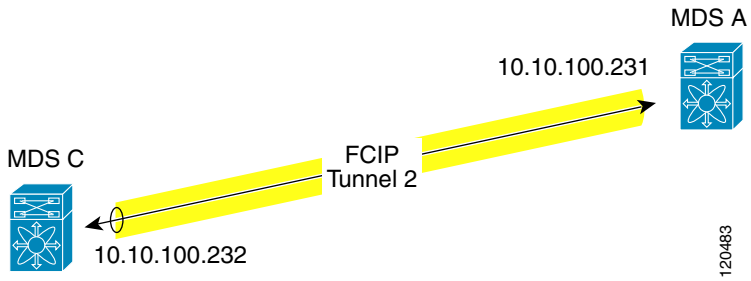
Example 7-19 Displays the Global SA Lifetime Values

```
switch# show crypto global domain ipsec security-association lifetime
Security Association Lifetime: 450 gigabytes/3600 seconds
```

Sample FCIP Configuration

Figure 7-7 focuses on implementing IPsec for one FCIP link (Tunnel 2). Tunnel 2 carries encrypted data between MDS A and MDS C.

Figure 7-7 IP Security Usage in an FCIP Scenario



To configure IPsec for the FCIP scenario shown in Figure 7-7, follow these steps:

Step 1 Enable IKE and IPsec in Switch MDS A.

```
sw10.1.1.100# conf t
sw10.1.1.100(config)# feature crypto ike
sw10.1.1.100(config)# feature crypto ipsec
```

Step 2 Configure IKE in Switch MDS A.

```
sw10.1.1.100(config)# crypto ike domain ipsec
sw10.1.1.100(config-ike-ipsec)# key ctct address 10.10.100.232
sw10.1.1.100(config-ike-ipsec)# policy 1
sw10.1.1.100(config-ike-ipsec-policy)# encryption 3des
sw10.1.1.100(config-ike-ipsec-policy)# hash md5
sw10.1.1.100(config-ike-ipsec-policy)# end
sw10.1.1.100#
```

Step 3 Configure the ACLs in Switch MDS A.

```
sw10.1.1.100# conf t
sw10.1.1.100(config)# ip access-list acl1 permit tcp 10.10.100.231 0.0.0.0 10.10.100.232
0.0.0.0
```

Step 4 Configure the transform set in Switch MDS A.

```
sw10.1.1.100(config)# crypto transform-set domain ipsec tfs-02 esp-aes 128 esp-sha1-hmac
```

Step 5 Configure the crypto map in Switch MDS A.

```
sw10.1.1.100(config)# crypto map domain ipsec cmap-01 1
sw10.1.1.100(config-crypto-map-ip)# match address acl1
sw10.1.1.100(config-crypto-map-ip)# set peer 10.10.100.232
sw10.1.1.100(config-crypto-map-ip)# set transform-set tfs-02
sw10.1.1.100(config-crypto-map-ip)# set security-association lifetime seconds 3600
sw10.1.1.100(config-crypto-map-ip)# set security-association lifetime gigabytes 3000
sw10.1.1.100(config-crypto-map-ip)# set pfs group5
sw10.1.1.100(config-crypto-map-ip)# end
sw10.1.1.100#
```

Step 6 Bind the interface to the crypto map set in Switch MDS A.

```
sw10.1.1.100# conf t
sw10.1.1.100(config)# int gigabitethernet 7/1
sw10.1.1.100(config-if)# ip addr 10.10.100.231 255.255.255.0
sw10.1.1.100(config-if)# crypto map domain ipsec cmap-01
sw10.1.1.100(config-if)# no shut
sw10.1.1.100(config-if)# exit
sw10.1.1.100(config)#
```

Step 7 Configure FCIP in Switch MDS A.

```
sw10.1.1.100(config)# feature fcip
sw10.1.1.100(config)# fcip profile 2
sw10.1.1.100(config-profile)# ip address 10.10.100.231
sw10.1.1.100(config-profile)# int fcip 2
sw10.1.1.100(config-if)# peer-info ipaddr 10.10.100.232
sw10.1.1.100(config-if)# use-profile 2
sw10.1.1.100(config-if)# no shut
sw10.1.1.100(config-if)# end
sw10.1.1.100#
```

Step 8 Verify the configuration in Switch MDS A.

```
sw10.1.1.100# show crypto global domain ipsec security-association lifetime
Security Association Lifetime: 4500 megabytes/3600 seconds

sw10.1.1.100# show crypto map domain ipsec
Crypto Map "cmap-01" 1 ipsec
        Peer = 10.10.100.232
        IP ACL = acl1
            permit ip 10.10.100.231 255.255.255.255 10.10.100.232 255.255.255.255
        Transform-sets: tfs-02,
        Security Association Lifetime: 3000 gigabytes/3600 seconds
        PFS (Y/N): Y
```

```
        PFS Group: group5
Interface using crypto map set cmap-01:
    GigabitEthernet7/1

sw10.1.1.100# show crypto transform-set domain ipsec
Transform set: tfs-02 {esp-aes 128 esp-sha1-hmac}
    will negotiate {tunnel}

sw10.1.1.100# show crypto spd domain ipsec
Policy Database for interface: GigabitEthernet7/1, direction: Both
#   0:      deny  udp any port eq 500 any
#   1:      deny  udp any any port eq 500
#   2:      permit ip 10.10.100.231 255.255.255.255 10.10.100.232 255.255.255.255
#  63:      deny  ip any any

sw10.1.1.100# show crypto ike domain ipsec
keepalive 3600

sw10.1.1.100# show crypto ike domain ipsec key
key ctct address 10.10.100.232

sw10.1.1.100# show crypto ike domain ipsec policy
Priority 1, auth pre-shared, lifetime 86300 secs, encryption 3des, hash md5, DH group 1
```

Step 9 Enable IKE and IPsec in Switch MDS C.

```
sw11.1.1.100# conf t
sw11.1.1.100(config)# feature crypto ike
sw11.1.1.100(config)# feature crypto ipsec
```

Step 10 Configure IKE in Switch MDS C.

```
sw11.1.1.100(config)# crypto ike domain ipsec
sw11.1.1.100(config-ike-ipsec)# key ctct address 10.10.100.231
sw11.1.1.100(config-ike-ipsec)# policy 1
sw11.1.1.100(config-ike-ipsec-policy)# encryption 3des
sw11.1.1.100(config-ike-ipsec-policy)# hash md5
sw11.1.1.100(config-ike-ipsec-policy)# exit
sw11.1.1.100(config-ike-ipsec)# end
sw11.1.1.100#
```

Step 11 Configure the ACLs in Switch MDS C.

```
sw11.1.1.100# conf t
sw11.1.1.100(config)# ip access-list acl1 permit ip 10.10.100.232 0.0.0.0 10.10.100.231
0.0.0.0
```

Step 12 Configure the transform set in Switch MDS C.

```
sw11.1.1.100(config)# crypto transform-set domain ipsec tfs-02 esp-aes 128 esp-sha1-hmac
```

Step 13 Configure the crypto map in Switch MDS C.

```
sw11.1.1.100(config)# crypto map domain ipsec cmap-01 1
sw11.1.1.100(config-crypto-map-ip)# match address acl1
sw11.1.1.100(config-crypto-map-ip)# set peer 10.10.100.231
sw11.1.1.100(config-crypto-map-ip)# set transform-set tfs-02
sw11.1.1.100(config-crypto-map-ip)# set security-association lifetime seconds 3600
sw11.1.1.100(config-crypto-map-ip)# set security-association lifetime gigabytes 3000
sw11.1.1.100(config-crypto-map-ip)# set pfs group5
sw11.1.1.100(config-crypto-map-ip)# exit
sw11.1.1.100(config)#
```

Step 14 Bind the interface to the crypto map set in Switch MDS C.

```
sw11.1.1.100(config)# int gigabitethernet 1/2
sw11.1.1.100(config-if)# ip addr 10.10.100.232 255.255.255.0
sw11.1.1.100(config-if)# crypto map domain ipsec cmap-01
sw11.1.1.100(config-if)# no shut
sw11.1.1.100(config-if)# exit
sw11.1.1.100(config)#
```

Step 15 Configure FCIP in Switch MDS C.

```
sw11.1.1.100(config)# feature fcip
sw11.1.1.100(config)# fcip profile 2
sw11.1.1.100(config-profile)# ip address 10.10.100.232
sw11.1.1.100(config-profile)# int fcip 2
sw11.1.1.100(config-if)# peer-info ipaddr 10.10.100.231
sw11.1.1.100(config-if)# use-profile 2
sw11.1.1.100(config-if)# no shut
sw11.1.1.100(config-if)# exit
sw11.1.1.100(config)# exit
```

Step 16 Verify the configuration in Switch MDS C.

```
sw11.1.1.100# show crypto global domain ipsec security-association lifetime
Security Association Lifetime: 4500 megabytes/3600 seconds

sw11.1.1.100# show crypto map domain ipsec
Crypto Map "cmap-01" 1 ipsec
        Peer = 10.10.100.231
        IP ACL = acl1
            permit ip 10.10.100.232 255.255.255.255 10.10.100.231 255.255.255.255
        Transform-sets: tfs-02,
        Security Association Lifetime: 3000 gigabytes/3600 seconds
        PFS (Y/N): Y
         PFS Group: group5
Interface using crypto map set cmap-01:
    GigabitEthernet1/2

sw11.1.1.100# show crypto spd domain ipsec
Policy Database for interface: GigabitEthernet1/2, direction: Both
#   0:      deny  udp any port eq 500 any
#   1:      deny  udp any any port eq 500
#   2:      permit ip 10.10.100.232 255.255.255.255 10.10.100.231 255.255.255.255
#  63:      deny  ip any any

sw11.1.1.100# show crypto sad domain ipsec
interface: GigabitEthernet1/2
    Crypto map tag: cmap-01, local addr. 10.10.100.232
    protected network:
    local  ident (addr/mask): (10.10.100.232/255.255.255.255)
    remote  ident (addr/mask): (10.10.100.231/255.255.255.255)
    current_peer: 10.10.100.231
      local crypto endpt.: 10.10.100.232, remote crypto endpt.: 10.10.100.231
      mode: tunnel, crypto algo: esp-3des, auth algo: esp-md5-hmac
     current outbound spi: 0x38f96001 (955867137), index: 29
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 3221225472000
     current inbound spi: 0x900b011 (151040017), index: 16
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 3221225472000

sw11.1.1.100# show crypto transform-set domain ipsec
Transform set: tfs-02 {esp-aes 128 esp-sha1-hmac}
    will negotiate {tunnel}

sw11.1.1.100# show crypto ike domain ipsec
```

```
keepalive 3600

sw11.1.1.100# show crypto ike domain ipsec key

key ctct address 10.10.100.231

sw11.1.1.100# show crypto ike domain ipsec policy
Priority 1, auth pre-shared, lifetime 86300 secs, encryption 3des, hash md5, DH
group 1

sw11.1.1.100# show crypto ike domain ipsec sa
Tunn    Local Addr          Remote Addr         Encr    Hash    Auth Method     Lifetime
------------------------------------------------------------------------------------------
1*      10.10.100.232[500]  10.10.100.231[500]  3des    md5     preshared key   86300
------------------------------------------------------------------------------------------
NOTE: tunnel id ended with * indicates an IKEv1 tunnel
```

Step 17 Verify the configuration in Switch MDS A.

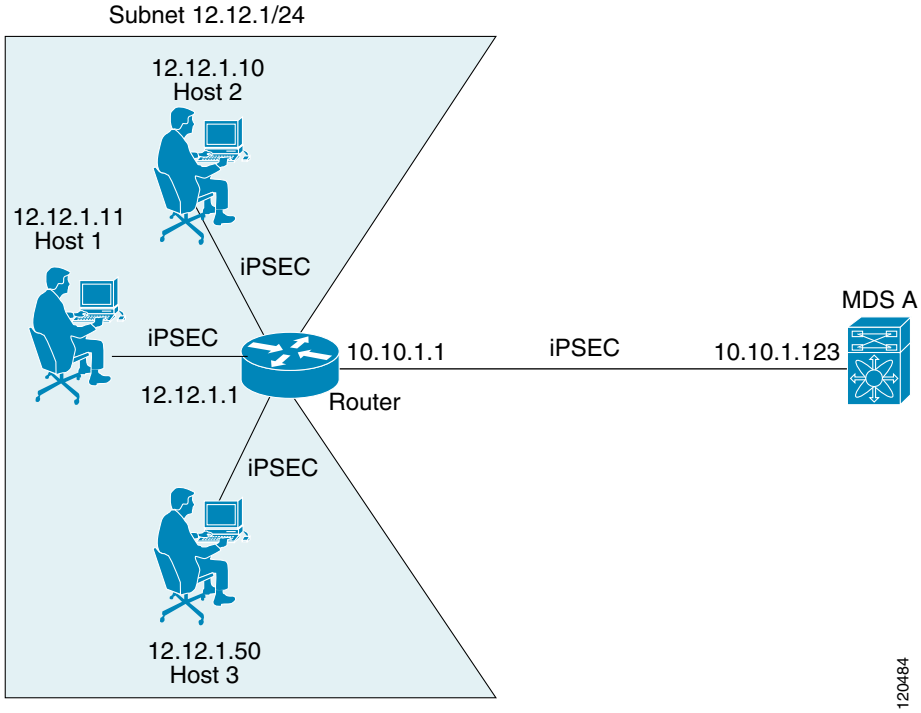
```
sw10.1.1.100# show crypto sad domain ipsec
interface: GigabitEthernet7/1
    Crypto map tag: cmap-01, local addr. 10.10.100.231
    protected network:
    local  ident (addr/mask): (10.10.100.231/255.255.255.255)
    remote  ident (addr/mask): (10.10.100.232/255.255.255.255)
    current_peer: 10.10.100.232
      local crypto endpt.: 10.10.100.231, remote crypto endpt.: 10.10.100.232
      mode: tunnel, crypto algo: esp-3des, auth algo: esp-md5-hmac
     current outbound spi: 0x900b01e (151040030), index: 10
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 3221225472000
     current inbound spi: 0x38fe700e (956198926), index: 13
      lifetimes in seconds:: 3600
      lifetimes in bytes:: 3221225472000

sw10.1.1.100# show crypto ike domain ipsec sa
Tunn Local Addr          Remote Addr         Encr  Hash  Auth Method   Lifetime
------------------------------------------------------------------------------
   1 10.10.100.231[500]  10.10.100.232[500]  3des  md5   preshared key    86300
```

You have now configured IPsec in both switches MDS A and MDS C.

Sample iSCSI Configuration

Figure 7-8 focuses on the iSCSI session between MDS A and the hosts in subnet 12.12.1/24. Using the **auto-peer** option, when any host from the subnet 12.12.1.0/24 tries to connect to the MDS switch's Gigabit Ethernet port 7/1, an SA is created between the hosts and the MDS switch. With auto-peer, only one crypto map is necessary to create SAs for all the hosts in the same subnet. Without auto-peer, you need one crypto map entry per host.

Figure 7-8 iSCSI with End-to-End IPsec



To configure IPsec for the iSCSI scenario shown in Figure 7-8, follow these steps:

Step 1 Configure the ACLs in Switch MDS A.

```
sw10.1.1.100# conf t
sw10.1.1.100(config)# ip access-list acl1 permit tcp 10.10.1.0 0.0.0.255 range port 3260
3260 12.12.1.0 0.0.0.255
```

Step 2 Configure the transform set in Switch MDS A.

```
sw10.1.1.100(config)# crypto transform-set domain ipsec tfs-01 esp-3des esp-md5-hmac
```

Step 3 Configure the crypto map in Switch MDS A.

```
sw10.1.1.100(config)# crypto map domain ipsec cmap-01 1
sw10.1.1.100(config-crypto-map-ip)# match address acl1
sw10.1.1.100(config-crypto-map-ip)# set peer auto-peer
sw10.1.1.100(config-crypto-map-ip)# set transform-set tfs-01
sw10.1.1.100(config-crypto-map-ip)# end
sw10.1.1.100#
```

Step 4 Bind the interface to the crypto map set in Switch MDS A.

```
sw10.1.1.100# conf t
sw10.1.1.100(config)# int gigabitethernet 7/1
sw10.1.1.100(config-if)# ip address 10.10.1.123 255.255.255.0
sw10.1.1.100(config-if)# crypto map domain ipsec cmap-01
sw10.1.1.100(config-if)# no shut
sw10.1.1.100(config-if)# end
sw10.1.1.100#
```

You have now configured IPsec in MDS A using the Cisco MDS IPsec and iSCSI features.

Default Settings

Table 7-4 lists the default settings for IKE parameters.

Table 7-4 Default IKE Parameters

Parameters	Default
IKE	Disabled.
IKE version	IKE version 2.
IKE encryption algorithm	3DES.
IKE hash algorithm	SHA.
IKE authentication method	Preshared keys.
IKE DH group identifier	Group 1.
IKE lifetime association	86,400 seconds (equals 24 hours).
IKE keepalive time for each peer (v2)	3,600 seconds (equals 1 hour).

Table 7-5 lists the default settings for IPsec parameters.

Table 7-5 Default IPsec Parameters

Parameters	Default
IPsec	Disabled.
Applying IPsec to the traffic.	Deny—allowing clear text.
IPsec PFS	Disabled.
IPsec global lifetime (traffic-volume)	450 Gigabytes.
IPsec global lifetime (time)	3,600 seconds (one hour).