

IPsec and ISAKMP

Released: April 24, 2014
Updated: June 26, 2014

This chapter describes how to configure Internet Protocol Security (IPsec) and the Internet Security Association and Key Management Protocol (ISAKMP) standards to build Virtual Private Networks (VPNs).

- Information About Tunneling, IPsec, and ISAKMP, page 1-1
- Licensing Requirements for Remote Access IPsec VPNs, page 1-3
- Guidelines and Limitations, page 1-6
- Configuring ISAKMP, page 1-7
- Configuring Certificate Group Matching for IKEv1, page 1-15
- Configuring IPsec, page 1-17
- Clearing Security Associations, page 1-36
- Clearing Crypto Map Configurations, page 1-36
- Supporting the Nokia VPN Client, page 1-37

Information About Tunneling, IPsec, and ISAKMP

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

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

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

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

IPsec Overview

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

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

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

Configuration for site to site tasks is performed in both single context mode and multiple context mode.



Note Multiple context mode only applies to IKEv2 and IKEv1 site to site and does not apply to AnyConnect, clientless SSL VPN, the legacy Cisco VPN client, the Apple native VPN client, the Microsoft native VPN client, or cTCP for IKEv1 IPsec.

ISAKMP and IKE Overview

ISAKMP is the negotiation protocol that lets two hosts agree on how to build an IPsec security association (SA). It provides a common framework for agreeing on the format of SA attributes. This security association includes negotiating with the peer about the SA and modifying or deleting the SA. ISAKMP separates negotiation into two phases: Phase 1 and Phase 2. Phase 1 creates the first tunnel, which protects later ISAKMP negotiation messages. Phase 2 creates the tunnel that protects data.

IKE uses ISAKMP to set up the SA for IPsec to use. IKE creates the cryptographic keys used to authenticate peers.

The ASA supports IKEv1 for connections from the legacy Cisco VPN client, and IKEv2 for the AnyConnect VPN client.

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

- The authentication type required of the IKEv1 peer, either RSA signature using certificates or preshared key (PSK).
- An encryption method to protect the data and ensure privacy.
- A Hashed Message Authentication Codes (HMAC) method to ensure the identity of the sender, and to ensure that the message has not been modified in transit.
- A Diffie-Hellman group to determine the strength of the encryption-key-determination algorithm. The ASA uses this algorithm to derive the encryption and hash keys.

- For IKEv2, a separate pseudo-random function (PRF) used as the algorithm to derive keying material and hashing operations required for the IKEv2 tunnel encryption and so on.
- A limit to the time the ASA uses an encryption key before replacing it.

With IKEv1 policies, you set one value for each parameter. For IKEv2, you can configure multiple encryption and authentication types, and multiple integrity algorithms for a single policy. The ASA orders the settings from the most secure to the least secure and negotiates with the peer using that order. This ordering allows you to potentially send a single proposal to convey all the allowed transforms instead of sending each allowed combination as with IKEv1.

Licensing Requirements for Remote Access IPsec VPNs

The following table shows the licensing requirements for this feature:

Note This feature is not available on No Payload Encryption models.w

Model	License Requirement[1]
ASA 5505	• IPsec remote access VPN using IKEv2 (use one of the following): – AnyConnect Premium license: Base license and Security Plus license: 2 sessions. *Optional permanent or time-based licenses: 10 or 25 sessions.* *Shared licenses are not supported.*[2] – AnyConnect Essentials license[3]: 25 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: – Base license: 10 sessions. – Security Plus license: 25 sessions.
ASA 5512-X	• IPsec remote access VPN using IKEv2 (use one of the following): – AnyConnect Premium license: Base license and Security Plus license: 2 sessions. *Optional permanent or time-based licenses: 10, 25, 50, 100, or 250 sessions.* *Optional Shared licenses*[2]*: Participant or Server. For the Server license, 500-50,000 in increments of 500 and 50,000-545,000 in increments of 1000.* – AnyConnect Essentials license[3]: 250 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: Base license and Security Plus license: 250 sessions.

Model	License Requirement[1]
ASA 5515-X	• IPsec remote access VPN using IKEv2 (use one of the following): – AnyConnect Premium license: Base license: 2 sessions. *Optional permanent or time-based licenses: 10, 25, 50, 100, or 250 sessions.* *Optional Shared licenses[2]: Participant or Server. For the Server license, 500-50,000 in increments of 500 and 50,000-545,000 in increments of 1000.* – AnyConnect Essentials license[3]: 250 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: Base license: 250 sessions.
ASA 5525-X	• IPsec remote access VPN using IKEv2 (use one of the following): – AnyConnect Premium license: Base license: 2 sessions. *Optional permanent or time-based licenses: 10, 25, 50, 100, 250, 500, or 750 sessions.* *Optional Shared licenses[2]: Participant or Server. For the Server license, 500-50,000 in increments of 500 and 50,000-545,000 in increments of 1000.* – AnyConnect Essentials license[3]: 750 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: Base license: 750 sessions.
ASA 5545-X	• IPsec remote access VPN using IKEv2 (use one of the following): – AnyConnect Premium license: Base license: 2 sessions. *Optional permanent or time-based licenses: 10, 25, 50, 100, 250, 500, 750, 1000, or 2500 sessions.* *Optional Shared licenses[2]: Participant or Server. For the Server license, 500-50,000 in increments of 500 and 50,000-545,000 in increments of 1000.* – AnyConnect Essentials license[3]: 2500 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: Base license: 2500 sessions.
ASA 5555-X	• IPsec remote access VPN using IKEv2 (use one of the following): – AnyConnect Premium license: Base license: 2 sessions. *Optional permanent or time-based licenses: 10, 25, 50, 100, 250, 500, 750, 1000, 2500, or 5000 sessions.* *Optional Shared licenses[2]: Participant or Server. For the Server license, 500-50,000 in increments of 500 and 50,000-545,000 in increments of 1000.* – AnyConnect Essentials license[3]: 5000 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: Base license: 5000 sessions.

Model	License Requirement[1]
ASA 5585-X with SSP-10	• IPsec remote access VPN using IKEv2 (use one of the following): – AnyConnect Premium license: Base license: 2 sessions. *Optional permanent or time-based licenses: 10, 25, 50, 100, 250, 500, 750, 1000, 2500, or 5000 sessions.* *Optional Shared licenses[2]: Participant or Server. For the Server license, 500-50,000 in increments of 500 and 50,000-545,000 in increments of 1000.* – AnyConnect Essentials license[3]: 5000 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: Base license: 5000 sessions.
ASA 5585-X with SSP-20, -40, and -60	• IPsec remote access VPN using IKEv2 (use one of the following): – AnyConnect Premium license: Base license: 2 sessions. *Optional permanent or time-based licenses: 10, 25, 50, 100, 250, 500, 750, 1000, 2500, 5000, or 10000 sessions.* *Optional Shared licenses[2]: Participant or Server. For the Server license, 500-50,000 in increments of 500 and 50,000-545,000 in increments of 1000.* – AnyConnect Essentials license[3]: 10000 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: Base license: 10000 sessions.
ASASM	• IPsec remote access VPN using IKEv2 (use one of the following): – AnyConnect Premium license: Base license: 2 sessions. *Optional permanent or time-based licenses: 10, 25, 50, 100, 250, 500, 750, 1000, 2500, 5000, or 10000 sessions.* *Optional Shared licenses[2]: Participant or Server. For the Server license, 500-50,000 in increments of 500 and 50,000-545,000 in increments of 1000.* – AnyConnect Essentials license[3]: 10000 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: Base license: 10000 sessions.

Model	License Requirement[1]
ASAv with 1 Virtual CPU	• IPsec remote access VPN using IKEv2: – Standard license: 2 sessions. – Premium license: 250 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: Standard and Premium licenses: 250 sessions.
ASAv with 4 Virtual CPUs	• IPsec remote access VPN using IKEv2: – Standard license: 2 sessions. – Premium license: 750 sessions. • IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2: Standard and Premium licenses: 750 sessions.

1. The maximum combined VPN sessions of *all* types cannot exceed the maximum sessions shown in this table. For the ASA 5505, the maximum combined sessions is 10 for the Base license, and 25 for the Security Plus license.
2. A shared license lets the ASA act as a shared license server for multiple client ASAs. The shared license pool is large, but the maximum number of sessions used by each individual ASA cannot exceed the maximum number listed for permanent licenses.
3. The AnyConnect Essentials license enables AnyConnect VPN client access to the ASA. This license does not support browser-based SSL VPN access or Cisco Secure Desktop. For these features, activate an AnyConnect Premium license instead of the AnyConnect Essentials license.

   Note: With the AnyConnect Essentials license, VPN users can use a Web browser to log in, and download and start (WebLaunch) the AnyConnect client.

   The AnyConnect client software offers the same set of client features, whether it is enabled by this license or an AnyConnect Premium SSL VPN Edition license.

   The AnyConnect Essentials license cannot be active at the same time as the following licenses on a given ASA: AnyConnect Premium license (all types) or the Advanced Endpoint Assessment license. You can, however, run AnyConnect Essentials and AnyConnect Premium licenses on different ASAs in the same network.

   By default, the ASA uses the AnyConnect Essentials license, but you can disable it to use other licenses by using the **webvpn**, and then the **no anyconnect-essentials** command.

   For a detailed list of the features supported by the AnyConnect Essentials license and AnyConnect Premium license, see *AnyConnect Secure Mobility Client Features, Licenses, and OSs*:

   http://www.cisco.com/en/US/products/ps10884/products_feature_guides_list.html

Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

Context Mode Guidelines

Supported in single or multiple context mode.

Firewall Mode Guidelines

Supported in routed firewall mode only. Does not support transparent firewall mode.

Failover Guidelines

IPsec VPN sessions are replicated in Active/Standby failover configurations only.

IPv6 Guidelines

Does not support IPv6.

Configuring ISAKMP

This section describes the Internet Security Association and Key Management Protocol (ISAKMP) and the Internet Key Exchange (IKE) protocol.

Configuring IKEv1 and IKEv2 Policies

To create an IKE policy, enter the **crypto *ikev1* | *ikev2* policy** command from global configuration mode in either single or multiple context mode. The prompt displays IKE policy configuration mode. For example:

```
hostname(config)# crypto ikev1 policy 1
hostname(config-ikev1-policy)#
```

After creating the policy, you can specify the settings for the policy.

Table 1-1 and Table 1-2 provide information about the IKEv1 and IKEv2 policy keywords and their values.

Table 1-1 IKEv1 Policy Keywords for CLI Commands

Command	Keyword	Meaning	Description
authentication	rsa-sig	A digital certificate with keys generated by the RSA signatures algorithm	Specifies the authentication method the ASA uses to establish the identity of each IPsec peer.
	crack	Challenge/Response for Authenticated Cryptographic Keys	CRACK provides strong mutual authentication when the client authenticates using a legacy method such as RADIUS, and the server uses public key authentication.
	pre-share (default)	Preshared keys	Preshared keys do not scale well with a growing network but are easier to set up in a small network.
encryption	**des**	56-bit DES-CBC	Specifies the symmetric encryption algorithm that protects data transmitted between two IPsec peers. The default is 168-bit Triple DES.
	3des (default)	168-bit Triple DES	
hash	**sha** (default)	SHA-1 (HMAC variant)	Specifies the hash algorithm used to ensure data integrity. It ensures that a packet comes from where it says it comes from and that it has not been modified in transit.
	md5	MD5 (HMAC variant)	The default is SHA-1. MD5 has a smaller digest and is considered to be slightly faster than SHA-1. A successful (but extremely difficult) attack against MD5 has occurred; however, the HMAC variant IKE uses prevents this attack.

Table 1-1 IKEv1 Policy Keywords for CLI Commands (continued)

Command	Keyword	Meaning	Description
group	1 **2** (default) 5	Group 1 (768-bit) Group 2 (1024-bit) Group 5 (1536-bit)	Specifies the Diffie-Hellman group identifier, which the two IPsec peers use to derive a shared secret without transmitting it to each other. The lower the Diffie-Hellman group number, the less CPU time it requires to execute. The higher the Diffie-Hellman group number, the greater the security. AES support is available on security appliances licensed for VPN-3DES only. To support the large key sizes required by AES, ISAKMP negotiation should use Diffie-Hellman (DH) Group 5.
lifetime	integer value (86400 = default)	120 to 2147483647 seconds	Specifies the SA lifetime. The default is 86,400 seconds or 24 hours. As a general rule, a shorter lifetime provides more secure ISAKMP negotiations (up to a point). However, with shorter lifetimes, the ASA sets up future IPsec SAs more quickly.

Table 1-2 IKEv2 Policy Keywords for CLI Commands

Command	Keyword	Meaning	Description
integrity	**sha** (default)	SHA-1 (HMAC variant)	Specifies the hash algorithm used to ensure data integrity. It ensures that a packet comes from where it says it comes from and that it has not been modified in transit.
	md5	MD5 (HMAC variant)	The default is SHA-1. MD5 has a smaller digest and is considered to be slightly faster than SHA-1. A successful (but extremely difficult) attack against MD5 has occurred; however, the HMAC variant IKE user prevents this attack.
	sha256	SHA 2, 256-bit digest	Specifies the Secure Hash Algorithm SHA 2 with the 256-bit digest.
	sha384	SHA 2, 384-bit digest	Specifies the Secure Hash Algorithm SHA 2 with the 384-bit digest.
	sha512	SHA 2, 512-bit digest	Specifies the Secure Hash Algorithm SHA 2 with the 512-bit digest.
	null		When AES-GCM is specified as the encryption algorithm, an administrator can choose null as the IKEv2 integrity algorithm.
encryption	**des**	56-bit DES-CBC	Specifies the symmetric encryption algorithm that protects data transmitted between two IPsec peers. The default is 168-bit Triple DES.
	3des (default)	168-bit Triple DES	
	aes aes-192 aes-256		The Advanced Encryption Standard supports key lengths of 128, 192, 256 bits.

Table 1-2 IKEv2 Policy Keywords for CLI Commands (continued)

Command	Keyword	Meaning	Description
	aes-gcm aes-gcm-192 aes-gcm-256 null	AES-GCM algorithm options to use for IKEv2 encryption	The Advanced Encryption Standard supports key lengths of 128, 192, 256 bits.
policy_index			Accesses the IKEv2 policy sub-mode.
prf	**sha** (default)	SHA-1 (HMAC variant)	Specifies the pseudo random function (PRF)—the algorithm used to generate keying material.
	md5	MD5 (HMAC variant)	The default is SHA-1. MD5 has a smaller digest and is considered to be slightly faster than SHA-1. A successful (but extremely difficult) attack against MD5 has occurred; however, the HMAC variant IKE uses prevents this attack.
	sha256	SHA 2, 256-bit digest	Specifies the Secure Hash Algorithm SHA 2 with the 256-bit digest.
	sha384	SHA 2, 384-bit digest	Specifies the Secure Hash Algorithm SHA 2 with the 384-bit digest.
	sha512	SHA 2, 512-bit digest	Specifies the Secure Hash Algorithm SHA 2 with the 512-bit digest.
priority			Extends the policy mode to support the additional IPsec V3 features and makes the AES-GCM and ECDH settings part of the Suite B support.
group	1	Group 1 (768-bit)	Specifies the Diffie-Hellman group identifier, which the two IPsec peers use to derive a shared secret without transmitting it to each other. The lower the Diffie-Hellman group number, the less CPU time it requires to execute. The higher the Diffie-Hellman group number, the greater the security. The AnyConnect client supports DH group 1, 2, and 5 in non-FIPS mode, and groups 2 and only in FIPS mode. AES support is available on security appliances licensed for VPN-3DES only. To support the large key sizes required by AES, ISAKMP negotiation should use Diffie-Hellman (DH) Group 5.
	2 (default)	Group 2 (1024-bit)	
	5	Group 5 (1536-bit)	
	14 19 20 21 24		
lifetime	integer value (86400 = default)	120 to 2147483647 seconds	Specifies the SA lifetime. The default is 86,400 seconds or 24 hours. As a general rule, a shorter lifetime provides more secure ISAKMP negotiations (up to a point). However, with shorter lifetimes, the ASA sets up future IPsec SAs more quickly.

IKEv1 and IKEv2 each support a maximum of 20 IKE policies, each with a different set of values. Assign a unique priority to each policy that you create. The lower the priority number, the higher the priority.

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

A match exists when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values. For IKEv1, the remote peer policy must also specify a lifetime less than or equal to the lifetime in the policy the initiator sent. If the lifetimes are not identical, the ASA uses the shorter lifetime. For IKEv2 the lifetime is not negotiated but managed locally between each peer, making it possible to configure lifetime independently on each peer. If no acceptable match exists, IKE refuses negotiation and the SA is not established.

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

Note New ASA configurations do not have a default IKEv1 or IKEv2 policy.

To configure IKE policies, in global configuration mode, use the **crypto *ikev1* | *ikev2* policy** *priority* command to enter IKE policy configuration mode.

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

To enable and configure IKE, complete the following steps, using the IKEv1 examples as a guide:

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

Step 1 Enter IKEv1 policy configuration mode:

```
hostname(config)# crypto ikev1 policy 1
hostname(config-ikev1-policy)#
```

Step 2 Specify the encryption algorithm. The default is Triple DES. This example sets encryption to DES.

```
encryption [aes | aes-192 | aes-256 | des | 3des]
```

For example:

```
hostname(config-ikev1-policy)# encryption des
```

Step 3 Specify the hash algorithm. The default is SHA-1. This example configures MD5.

```
hash [md5 | sha]
```

For example:

```
hostname(config-ikev1-policy)# hash md5
```

Step 4 Specify the authentication method. The default is preshared keys. This example configures RSA signatures.

```
authentication [pre-share | crack | rsa-sig]
```

For example:

```
hostname(config-ikev1-policy)# authentication rsa-sig
```

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

```
group [1 | 2 | 5]
```

For example:

```
hostname(config-ikev1-policy)# group 5
```

Step 6 Specify the SA lifetime. This examples sets a lifetime of 4 hours (14400 seconds). The default is 86400 seconds (24 hours).

lifetime *seconds*

For example:

```
hostname(config-ikev1-policy)# lifetime 14400
```

Enabling IKE on the Outside Interface

You must enable IKE on the interface that terminates the VPN tunnel. Typically this is the outside, or public interface. To enable IKEv1 or IKEv2, use the **crypto** ***ikev1*** **|** ***ikev2*** **enable** *interface-name* command from global configuration mode in either single or multiple context mode.

For example:

```
hostname(config)# crypto ikev1 enable outside
```

Disabling IKEv1 Aggressive Mode

Phase 1 IKEv1 negotiations can use either main mode or aggressive mode. Both provide the same services, but aggressive mode requires only two exchanges between the peers totaling three messages, rather than three exchanges totaling six messages. Aggressive mode is faster, but does not provide identity protection for the communicating parties. Therefore, the peers must exchange identification information before establishing a secure SA. Aggressive mode is enabled by default.

- Main mode is slower, using more exchanges, but it protects the identities of the communicating peers.
- Aggressive mode is faster, but does not protect the identities of the peers.

To disable aggressive mode, enter the following command in either single or multiple context mode:

```
crypto ikev1 am-disable
```

For example:

```
hostname(config)# crypto ikev1 am-disable
```

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

```
hostname(config)# no crypto ikev1 am-disable
```

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

Determining an ID Method for IKEv1 and IKEv2 ISAKMP Peers

During ISAKMP Phase I negotiations, either IKEv1 or IKEv2, the peers must identify themselves to each other. You can choose the identification method from the following options.

Address	Uses the IP addresses of the hosts exchanging ISAKMP identity information.
Automatic	Determines ISAKMP negotiation by connection type: • IP address for preshared key. • Cert Distinguished Name for certificate authentication.
Hostname	Uses the fully qualified domain name of the hosts exchanging ISAKMP identity information (default). This name comprises the hostname and the domain name.
Key ID *key_id_string*	Specifies the string used by the remote peer to look up the preshared key.

The ASA uses the Phase I ID to send to the peer. This is true for all VPN scenarios except LAN-to-LAN IKEv1 connections in main mode that authenticate with preshared keys.

The default setting is auto.

To change the peer identification method, enter the following command in either single or multiple context mode:

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

For example, the following command sets the peer identification method to hostname:

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

Enabling IPsec over NAT-T

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

Note Due to a limitation of the AnyConnect client, you must enable NAT-T for the AnyConnect client to successfully connect using IKEv2. This requirement applies even if the client is not behind a NAT-T device.

With the exception of the home zone on the Cisco ASA 5505, the ASA can simultaneously support standard IPsec, IPsec over TCP, NAT-T, and IPsec over UDP, depending on the client with which it is exchanging data.

The following breakdown shows the connections with each option enabled.

Options	Enabled Feature	Client Position	Feature Used
Option 1	If NAT-T is enabled	and client is behind NAT, then	NAT-T is used
		and no NAT exists, then	Native IPsec (ESP) is used

Options	Enabled Feature	Client Position	Feature Used
		and client is behind NAT, then	IPsec over UDP is used
Option 2	If IPsec over UDP is enabled	and no NAT exists, then	IPsec over UDP is used
	If both NAT-T and	and client is behind NAT, then	NAT-T is used
Option 3	IPsec over UDP are enabled	and no NAT exists, then	IPsec over UDP is used

Note When IPsec over TCP is enabled, it takes precedence over all other connection methods.

When you enable NAT-T, the ASA automatically opens port 4500 on all IPsec-enabled interfaces.

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

- LAN-to-LAN
- Remote access

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

Using NAT-T

To use NAT-T, you must perform the following site-to-site steps in either single or multiple context mode:

Step 1 Enter the following command to enable IPsec over NAT-T globally on the ASA:

```
crypto isakmp nat-traversal natkeepalive
```

The range for the *natkeepalive* argument is 10 to 3600 seconds. The default is 20 seconds.

For example, enter the following command to enable NAT-T and set the keepalive value to one hour.

```
hostname(config)# crypto isakmp nat-traversal 3600
```

Step 2 Select the before-encryption option for the IPsec fragmentation policy by entering this command:

```
hostname(config)# crypto ipsec fragmentation before-encryption
```

This option lets traffic travel across NAT devices that do not support IP fragmentation. It does not impede the operation of NAT devices that do support IP fragmentation.

Enabling IPsec with IKEv1 over TCP

IPsec/IKEv1 over TCP enables a Cisco VPN client to operate in an environment in which standard ESP or IKEv1 cannot function or can function only with modification to existing firewall rules. IPsec over TCP encapsulates both the IKEv1 and IPsec protocols within a TCP-like packet and enables secure tunneling through both NAT and PAT devices and firewalls. This feature is disabled by default.

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

IPsec over TCP works with remote access clients. You enable it globally, and it works on all IKEv1-enabled interfaces. It is a client to the ASA feature only. It does not work for LAN-to-LAN connections.

The ASA can simultaneously support standard IPsec, IPsec over TCP, NAT-Traversal, and IPsec over UDP, depending on the client with which it is exchanging data. IPsec over TCP, if enabled, takes precedence over all other connection methods.

The VPN 3002 hardware client, which supports one tunnel at a time, can connect using standard IPsec, IPsec over TCP, NAT-Traversal, or IPsec over UDP.

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

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

The default port is 10000.

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

To enable IPsec over TCP for IKEv1 globally on the ASA, perform the following command in either single or multiple context mode:

```
crypto ikev1 ipsec-over-tcp [port port 1...port0]
```

This example enables IPsec over TCP on port 45:

```
hostname(config)# crypto ikev1 ipsec-over-tcp port 45
```

Waiting for Active Sessions to Terminate Before Rebooting

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

To enable waiting for all active sessions to voluntarily terminate before the ASA reboots, perform the following site-to-site task in either single or multiple context mode:

```
crypto isakmp reload-wait
```

For example:

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

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

Alerting Peers Before Disconnecting

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

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

Qualified clients and peers include the following:

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

To enable disconnect notification to IPsec peers, enter the **crypto isakmp disconnect-notify** command in either single or multiple context mode.

For example:

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

Configuring Certificate Group Matching for IKEv1

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

Note Certificate group matching applies to IKEv1 and IKEv2 LAN-to-LAN connections only. IKEv2 remote access connections support the pull-down group selection configured in the webvpn-attributes of the tunnel-group and webvpn configuration mode for certificate-group-map, and so on.

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

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

You must also configure a certificate group matching policy, specifying to match the group from the rules, or from the organizational unit (OU) field, or to use a default group for all certificate users. You can use any or all of these methods.

The following sections provide more information:

- Creating a Certificate Group Matching Rule and Policy, page 1-16
- Using the Tunnel-group-map default-group Command, page 1-17

Creating a Certificate Group Matching Rule and Policy

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

The syntax follows:

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

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

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

Be aware of the following:

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

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

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

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

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

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

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

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

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

Using the Tunnel-group-map default-group Command

This command specifies a default tunnel group to use when the configuration does not specify a tunnel group.

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

Configuring IPsec

This section provides background information about IPsec and describes the procedures required to configure the ASA when using IPsec to implement a VPN.

Understanding IPsec Tunnels

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

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

- IKEv1 transform sets or IKEv2 proposals
- Crypto maps
- ACLs
- Tunnel groups
- Prefragmentation policies

Understanding IKEv1 Transform Sets and IKEv2 Proposals

An IKEv1 transform set or an IKEv2 proposal is a combination of security protocols and algorithms that define how the ASA protects data. During IPsec SA negotiations, the peers must identify a transform set or proposal that is the same at both peers. The ASA then applies the matching transform set or proposal to create an SA that protects data flows in the ACL for that crypto map.

With IKEv1 transform sets, you set one value for each parameter. For IKEv2 proposals, you can configure multiple encryption and authentication types and multiple integrity algorithms for a single proposal. The ASA orders the settings from the most secure to the least secure and negotiates with the peer using that order. This allows you to potentially send a single proposal to convey all the allowed combinations instead of the need to send each allowed combination individually as with IKEv1.

The ASA tears down the tunnel if you change the definition of the transform set or proposal used to create its SA. See the Clearing Security Associations, page 1-36" for further information.

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

Defining Crypto Maps

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

- ACL to identify the packets that the IPsec connection permits and protects.
- Peer identification.
- Local address for the IPsec traffic. (See Applying Crypto Maps to Interfaces, page 1-27 for more details.)
- Up to 11 IKEv1 transform sets or IKEv2 proposals, with which to attempt to match the peer security settings.

A *crypto map set* consists of one or more crypto maps that have the same map name. You create a crypto map set when you create its first crypto map. The following site-to-site task creates or adds to a crypto map in either single or multiple context mode:

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

Use the access-list-name to specify the ACL ID, as a string or integer up to 241 characters in length.

Tip Use all capital letters to more easily identify the ACL ID in your configuration.

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

```
crypto map mymap 10 match address 101
```

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

```
[no] crypto map <map_name> <map_index> set pfs [group1 | group2 | group5 | group14 |
group19 | group20 | group21 | group24]
```

Specifies the ECDH group used for Perfect Forward Secrecy (FCS) for the cryptography map. Prevents you from onfiguring group14 and group24 options for a cryptography map (when using an IKEv1 policy).

```
[no]crypto map <name> <priority> set validate-icmp-errors
OR
```

```
[no]crypto dynamic-map <name> <priority> set validate-icmp-errors
```

Specifies whether incoming ICMP error messages are validated for the cryptography or dynamic cryptographyy map.

```
[no] crypto map <name> <priority> set df-bit [clear-df | copy-df | set-df}
OR
[no] crypto map dynamic-map <name> <priority> set df-bit [clear-df | copy-df | set-df]
```

Configures the existing do not fragment (DF) policy (at a security association level) for the cryptography or dynamic cryptography map.

- *clear-df*—Ignores the DF bit.
- *copy-df*—Maintains the DF bit.
- *set-df*—Sets and uses the DF bit.

```
[no] crypto map <name> <priority> set tfc-packets [burst <length | auto] [payload-size
<bytes | auto> [timeout <seconds | auto>
OR
[no] crypto dynamic-map <name> <priority> set tfc-packets [burst <length | auto]
[payload-size <bytes | auto> [timeout <seconds | auto>
```

An administrator can enable dummy Traffic Flow Confidentiality (TFC) packets at random lengths and intervals on an IPsec security association. You must have an IKEv2 IPsec proposal set before enabling TFC.

The ACL assigned to a crypto map consists of all of the ACEs that have the same ACL name, as shown in the following command syntax:

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

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

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

In the following example, the ASA applies the IPsec protections assigned to the crypto map to all traffic flowing from the 10.0.0.0 subnet to the 10.1.1.0 subnet:

```
access-list 101 permit ip 10.0.0.0 255.255.255.0 10.1.1.0 255.255.255.0
```

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

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

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

You can apply only one crypto map set to a single interface. Create more than one crypto map for a particular interface on the ASA if any of the following conditions exist:

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

For example, create a crypto map and assign an ACL to identify traffic between two subnets and assign one IKEv1 transform set or IKEv2 proposal. Create another crypto map with a different ACL to identify traffic between another two subnets and apply a transform set or proposal with different VPN parameters.

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

Each ACE contains a permit or deny statement. Table 1-3 explains the special meanings of permit and deny ACEs in ACLs applied to crypto maps.

Table 1-3 Special Meanings of Permit and Deny in Crypto ACLs Applied to Outbound Traffic

Result of Crypto Map Evaluation	Response
Match criterion in an ACE containing a permit statement	Halt further evaluation of the packet against the remaining ACEs in the crypto map set, and evaluate the packet security settings against those in the IKEv1 transform sets or IKEv2 proposals assigned to the crypto map. After matching the security settings to those in a transform set or proposal, the ASA applies the associated IPsec settings. Typically for outbound traffic, this means that it decrypts, authenticates, and routes the packet.
Match criterion in an ACE containing a deny statement	Interrupt further evaluation of the packet against the remaining ACEs in the crypto map under evaluation, and resume evaluation against the ACEs in the next crypto map, as determined by the next seq-num assigned to it.
Fail to match all tested permit ACEs in the crypto map set	Route the packet without encrypting it.

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

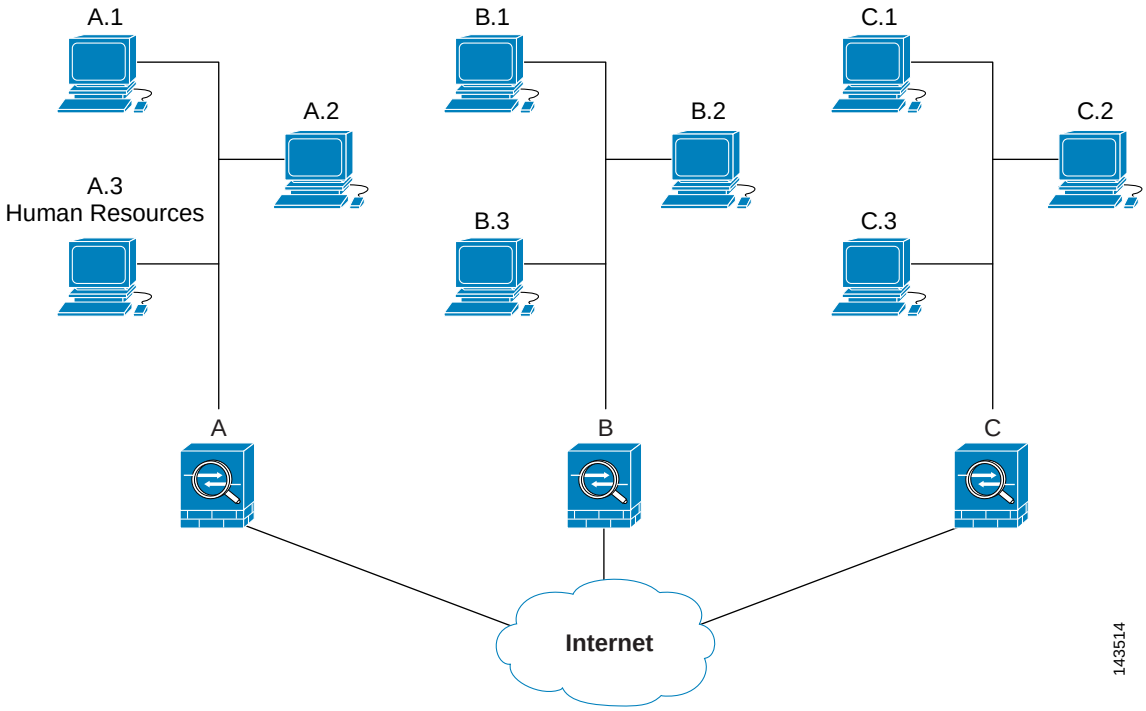
For an inbound, encrypted packet, the security appliance uses the source address and ESP SPI to determine the decryption parameters. After the security appliance decrypts the packet, it compares the inner header of the decrypted packet to the permit ACEs in the ACL associated with the packet SA. If the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

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

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

Figure 1-1 shows an example LAN-to-LAN network of ASAs.

Figure 1-1 Effect of Permit and Deny ACEs on Traffic (Conceptual Addresses)



The simple address notation shown in this figure and used in the following explanation is an abstraction. An example with real IP addresses follows the explanation.

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

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

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

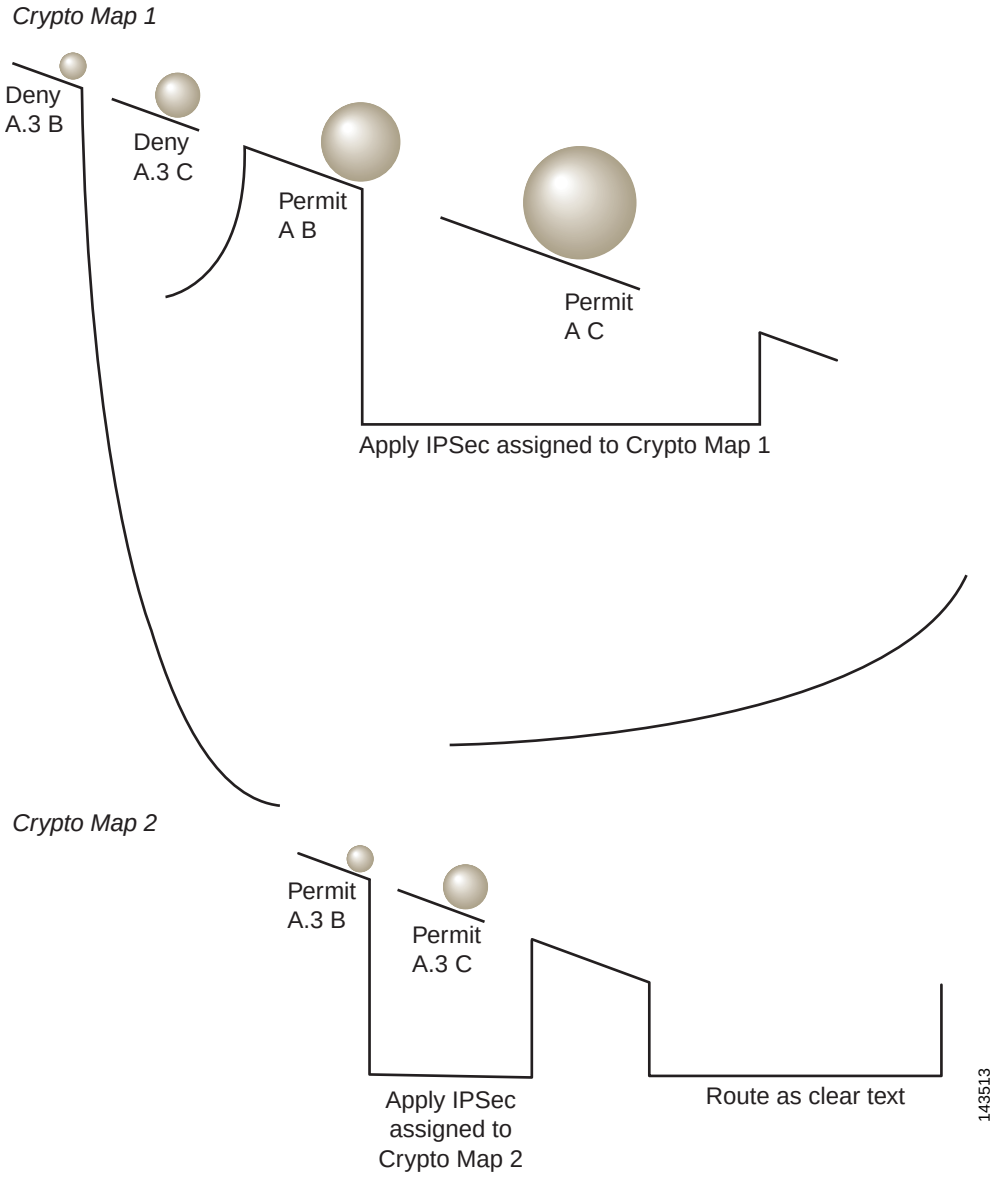
After creating the ACLs, you assign a transform set to each crypto map to apply the required IPsec to each matching packet.

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

Figure 1-2 shows the cascading ACLs created from the conceptual ACEs in this example. The meaning of each symbol in the figure follows.

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

Figure 1-2 Cascading ACLs in a Crypto Map Set



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

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

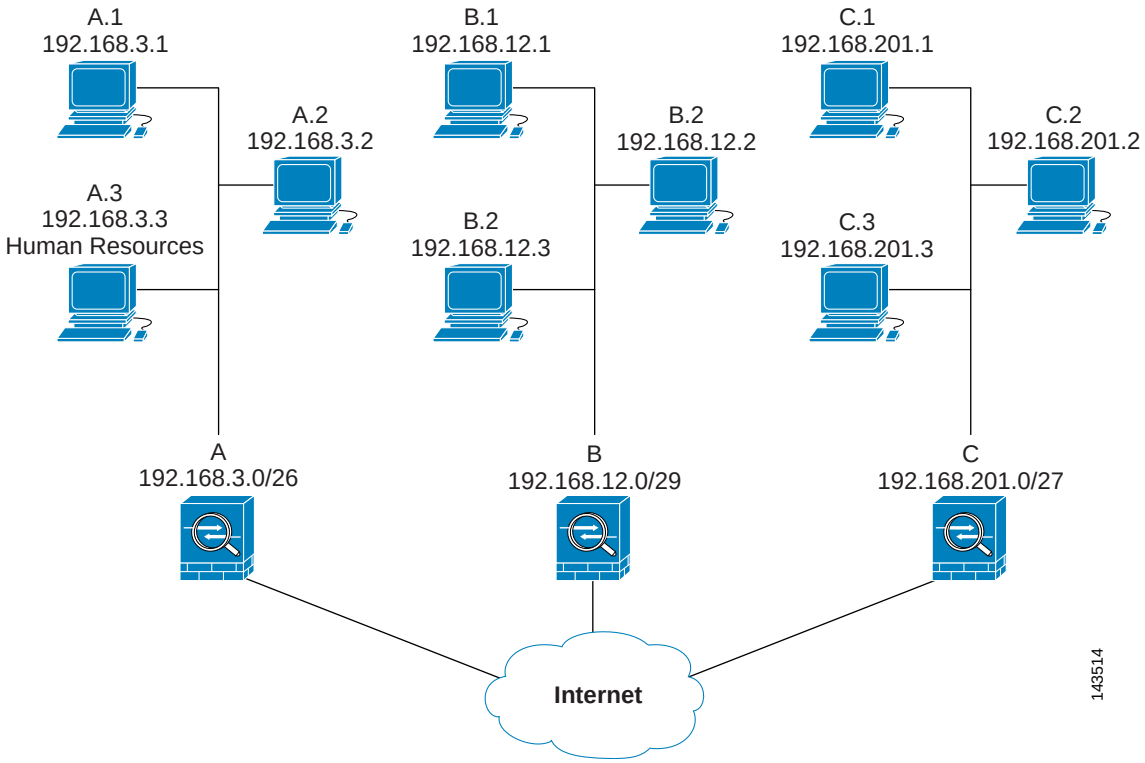
Table 1-4 shows the ACLs assigned to the crypto maps configured for all three ASAs in Figure 1-1.

Table 1-4 Example Permit and Deny Statements (Conceptual)

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

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

Figure 1-3 Effect of Permit and Deny ACEs on Traffic (Real Addresses)



The tables that follow combine the IP addresses shown in Figure 1-3 to the concepts shown in Table 1-4. The real ACEs shown in these tables ensure that all IPsec packets under evaluation within this network receive the proper IPsec settings.

Table 1-5 Example Permit and Deny Statements for Security Appliance A

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

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

Note By default, the ASA does not support IPsec traffic destined for the same interface from which it enters. Names for this type of traffic include U-turn, hub-and-spoke, and hairpinning. However, you can configure IPsec to support U-turn traffic by inserting an ACE to permit traffic to and from the network. For example, to support U-turn traffic on Security Appliance B, add a conceptual “permit B B” ACE to ACL1. The actual ACE would be as follows:

```
permit 192.168.12.0 255.255.255.248 192.168.12.0 255.255.255.248
```

Managing Public Key Infrastructure (PKI) Keys

You must set public key infrastructure (PKI) in order for an administrator to choose the Suite B ECDSA algorithms when generating or zeroing a keypair:

Prerequisites

If you are configuring a cryptography map to use an RSA or ECDSA trustpoint for authentication, you must first generate the key set. You can then create the trustpoint and reference it in the tunnel group configuration.

Detailed Steps

Step 1 Choose the Suite B ECDSA algorithm when generating a keypair:

```
crypto key generate [rsa [general-keys | label <name> | modules [512 | 768 | 1024 | 2048 |
4096 ] | noconfirm | usage-keys] | ecdsa [label <name> | elliptic-curve [256 | 384 | 521]
| noconfirm] ]
```

Step 2 Choose the Suite B ECDSA algorithem when zeroizing a keypair:

```
crypto key zeroize [rsa | ecdsa] [default | label <name> | noconfirm]
```

Configuring the Pool of Cryptographic Cores

You can change the allocation of the cryptographic cores on Symmetric Multi-Processing (SMP) platforms to give you better throughput performance for AnyConnect TLS/DTLS traffic. These changes can accelerate the SSL VPN datapath and provide customer-visible performance gains in AnyConnect, smart tunnels, and port forwarding. To configure the pool of cryptographic cores, perform the following steps.

Limitations

- Cryptographic core rebalancing is available on the following platforms:
  - 5585-X
  - 5545-X/5555-X
  - ASASM

Detailed Steps

Step 1 Configure the pool of cryptographic cores specifying one of three mutually exclusive options:

- balanced—Equally distributes cryptography hardware resources (Admin/SSL and IPsec cores).
- ipsec—Allocates cryptography hardware resources to favor IPsec (includes SRTP encrypted voice traffic).
- ssl—Allocates cryptography hardware resources to favor Admin/SSL.

```
asa1(config)# crypto engine ?

configure mode commands/options:
accelerator-bias
Specify how to allocate crypto accelerator processors

asa1(config)# crypto engine accelerator-bias ?
configure mode commands/options
balanced - Equally distribute crypto hardware resources
ipsec - Allocate crypto hardware resources to favor IPsec/Encrypted Voice (SRTP)
ssl - Allocate crypto hardware resources to favor SSL

asa1(config)# crypto engine accelerator-bias ssl
```

Applying Crypto Maps to Interfaces

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

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

Using Interface ACLs

By default, the ASA lets IPsec packets bypass interface ACLs. If you want to apply interface ACLs to IPsec traffic, use the **no** form of the **sysopt connection permit-vpn** command.

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

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

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

ACLs assigned to IPsec crypto maps have four primary functions:

- Select outbound traffic to be protected by IPsec (permit = protect).
- Trigger an ISAKMP negotiation for data travelling without an established SA.
- Process inbound traffic to filter out and discard traffic that should have been protected by IPsec.
- Determine whether to accept requests for IPsec SAs when processing IKE negotiation from the peer. (Negotiation applies only to **ipsec-isakmp crypto map** entries.) The peer must permit a data flow associated with an **ipsec-isakmp crypto map** command entry to ensure acceptance during negotiation.

Regardless of whether the traffic is inbound or outbound, the ASA evaluates traffic against the ACLs assigned to an interface. Follow these steps to assign IPsec to an interface:

Step 1 Create the ACLs to be used for IPsec.

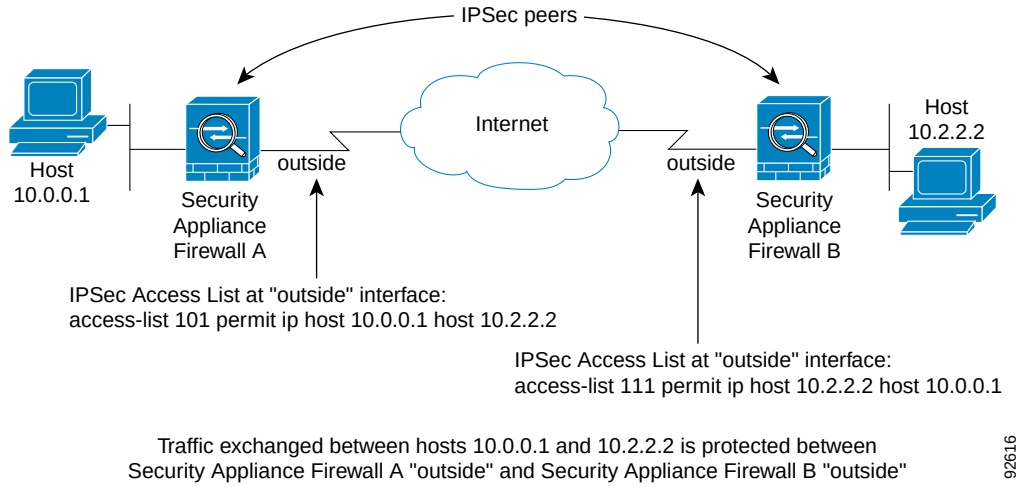
Step 2 Map the lists to one or more crypto maps, using the same crypto map name.

Step 3 Map the IKEv1 transform sets or IKEv2 proposals to the crypto maps to apply IPsec to the data flows.

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

In Figure 1-4, IPsec protection applies to traffic between Host 10.0.0.1 and Host 10.2.2.2 as the data exits the outside interface on Security Appliance A toward Host 10.2.2.2.

Figure 1-4 How Crypto ACLs Apply to IPsec



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

- source = host 10.0.0.1
- dest = host 10.2.2.2

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

- source = host 10.2.2.2
- dest = host 10.0.0.1

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

Note If you delete the only element in an ACL, the ASA also removes the associated crypto map.

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

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

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

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

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

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

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



Note Decrypted through traffic is permitted from the client despite having an access group on the outside interface, which calls a deny ip any any access-list, while **no sysopt connection permit-vpn** is configured.

Users who want to control access to the protected network via site-to-site or remote access VPN using the **no sysopt permit** command in conjunction with an access control list (ACL) on the outside interface are not successful.

In this situation, when management-access inside is enabled, the ACL is not applied, and users can still connect using SSH to the security appliance. Traffic to hosts on the inside network are blocked correctly by the ACL, but cannot block decrypted through traffic to the inside interface.

The **ssh** and **http** commands are of a higher priority than the ACLs. In other words, to deny SSH, Telnet, or ICMP traffic to the device from the VPN session, use **ssh**, **telnet** and **icmp** commands, which deny the IP local pool should be added.

Changing IPsec SA Lifetimes

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

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

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

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

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

Creating a Basic IPsec Configuration

You can create basic IPsec configurations with static or dynamic crypto maps.

To create a basic IPsec configuration using a static crypto map, perform the following steps:

Step 1 To create an ACL to define the traffic to protect, enter the following command:

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

For example:

```
hostname(config)# access-list 101 permit ip 10.0.0.0 255.255.255.0 10.1.1.0 255.255.255.0
```

The *access-list-name* specifies the ACL ID, as a string or integer up to 241 characters in length. The *destination-netmask* and *source-netmask* specifies an IPv4 network address and subnet mask. In this example, the **permit** keyword causes all traffic that matches the specified conditions to be protected by crypto.

Step 2 To configure an IKEv1 transform set that defines how to protect the traffic, enter the following command:

```
crypto ipsec ikev1 transform-set transform-set-name encryption [authentication]
```

Encryption specifies which encryption method protects IPsec data flows:

- esp-aes—Uses AES with a 128-bit key.
- esp-aes-192—Uses AES with a 192-bit key.
- esp-aes-256—Uses AES with a 256-bit key.\
- esp-des—Uses 56-bit DES-CBC.
- esp-3des—Uses triple DES algorithm.
- esp-null—No encryption.

Authentication specifies which encryption method to protect IPsec data flows:

- esp-md5-hmac—Uses the MD5/HMAC-128 as the hash algorithm.
- esp-sha-hmac—Uses the SHA/HMAC-160 as the hash algorithm.
- esp-none—No HMAC authentication.

For example:

```
hostname(config)# crypto ipsec ikev1 transform-set myset1 esp-des esp-sha-hmac
hostname(config)# crypto ipsec ikev1 transform-set myset2 esp-3des esp-sha-hmac
hostname(config)# crypto ipsec ikev1 transform-set aes_set esp-md5-hmac esp-aes-256
```

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

To configure an IKEv2 proposal that also defines how to protect the traffic, enter the **crypto ipsec ikev2 ipsec-proposal** command to create the proposal and enter the ipsec proposal configuration mode where you can specify multiple encryption and integrity types for the proposal:

```
crypto ipsec ikev2 ipsec-proposal [proposal tag]
```

Proposal tag is the name of the IKEv2 IPsec proposal, a string from 1 to 64 characters.

For example:

```
hostname(config)# crypto ipsec ikev2 ipsec-proposal secure
```

In this example, secure is the name of the proposal. Enter a protocol and encryption types:

```
hostname(config-ipsec-proposal)# protocol esp encryption 3des aes des
```

Conversely, the following command chooses which AES-GCM or AES-GMAC algorithm to use:

```
hostname(config-ipsec-proposal)# [no] protocol esp encryption [3des | aes | aes-192 |
aes-256 | aes-gcm | aes-gcm-192 | aes-gcm-256 | aes-gmac | aes-gmac-192 | aes-gmac-256 |
des | null]
```

If SHA-2 or null is chosen, you must choose which algorithm to use as an IPsec integrity algorithm. You must choose the null integriy algorithm if AES-GCM/GMAC is configured as the encryption algorithm:

```
hostname(config-ipsec-proposal)# [no] protocol esp integrity [md5 | sha-1 | sha-256 |
sha-384 | sha-512 | null]
```

Note You must choose the null integrity algorithm if AES-GCM/GMAC has been configured as the encryption algorithm. SHA-256 can be used for integrity and PRF to establish IKEv2 tunnels, but it can also be used for ESP integrity protection.

Step 3 (Optional) An administrator can enable path maximum transfer unit (PMTU) aging and set the interval at which the PMTU value is reset to its original value.

```
hostname(config-ipsec-proposal)# [no] crypto ipsec security-association pmtu-aging
<reset-interval>
```

Step 4 To create a crypto map, perform the following site-to-site steps using either single or multiple context mode:

a. Assign an ACL to a crypto map:

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

A crypto map set is a collection of crypto map entries, each with a different sequence number (*seq-num)* but the same *map name*. Use the *access-list-name* to specify the ACL ID, as a string or integer up to 241 characters in length. In the following example, mymap is the name of the crypto map set. The map set sequence number 10, which is used to rank multiple entries within one crypto map set. The lower the sequence number, the higher the priority.

```
crypto map mymap 10 match address 101
```

In this example, the ACL named 101 is assigned to crypto map mymap.

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

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

For example:

```
crypto map mymap 10 set peer 192.168.1.100
```

The ASA sets up an SA with the peer assigned the IP address 192.168.1.100. Specify multiple peers by repeating this command.

c. Specify which IKEv1 transform sets or IKEv2 proposals are allowed for this crypto map. List multiple transform sets or proposals in order of priority (highest priority first). You can specify up to 11 transform sets or proposals in a crypto map using either of these two commands:

```
crypto map map-name seq-num set ikev1 transform-set transform-set-name1
[transform-set-name2, …transform-set-name11]

crypto map map-name seq-num set ikev2 ipsec-proposal proposal-name1
[proposal-name2, … proposal-name11]
```

Proposal-name1 and *proposal-name11* specifies one or more names of the IPsec proposals for IKEv2. Each crypto map entry supports up to 11 proposals.

For example (for IKEv1):

```
crypto map mymap 10 set ikev1 transform-set myset1 myset2
```

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

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

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

Map-name specifies the name of the crypto map set. *Seq-num* specifies the number you assign to the crypto map entry.

For example:

```
crypto map mymap 10 set security-association lifetime seconds 2700
```

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

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

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

For example:

```
crypto map mymap 10 set pfs group2
```

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

Step 5 Apply a crypto map set to an interface for evaluating IPsec traffic:

```
crypto map map-name interface interface-name
```

Map-name specifies the name of the crypto map set. *Interface-name* specifies the name of the interface on which to enable or disable ISAKMP IKEv1 negotiation.

For example:

```
crypto map mymap interface outside
```

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

Using Dynamic Crypto Maps

A dynamic crypto map is a crypto map without all of the parameters configured. It acts as a policy template where the missing parameters are later dynamically learned, as the result of an IPsec negotiation, to match the peer requirements. The ASA applies a dynamic crypto map to let a peer negotiate a tunnel if its IP address is not already identified in a static crypto map. This occurs with the following types of peers:

- Peers with dynamically assigned public IP addresses.

  Both LAN-to-LAN and remote access peers can use DHCP to obtain a public IP address. The ASA uses this address only to initiate the tunnel.

- Peers with dynamically assigned private IP addresses.

Peers requesting remote access tunnels typically have private IP addresses assigned by the headend. Generally, LAN-to-LAN tunnels have a predetermined set of private networks that are used to configure static maps and therefore used to establish IPsec SAs.

As an administrator configuring static crypto maps, you might not know the IP addresses that are dynamically assigned (via DHCP or some other method), and you might not know the private IP addresses of other clients, regardless of how they were assigned. VPN clients typically do not have static IP addresses; they require a dynamic crypto map to allow IPsec negotiation to occur. For example, the headend assigns the IP address to a Cisco VPN client during IKE negotiation, which the client then uses to negotiate IPsec SAs.

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

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

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

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

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

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

Caution Do not assign module default routes for traffic to be tunneled to a ASA interface configured with a dynamic crypto map set. To identify the traffic that should be tunneled, add the ACLs to the dynamic crypto map. Use care to identify the proper address pools when configuring the ACLs associated with remote access tunnels. Use Reverse Route Injection to install routes only after the tunnel is up.

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

Follow these steps to create a crypto dynamic map entry using either single or multiple context mode:

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

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

This determines which traffic should be protected and not protected. *Dynamic-map-name* specifies the name of the crypto map entry that refers to a pre-existing dynamic crypto map. *Dynamic-seq-num* specifies the sequence number that corresponds to the dynamic crypto map entry.

For example:

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

In this example, ACL 101 is assigned to dynamic crypto map dyn1. The map sequence number is 10.

Step 2 Specify which IKEv1 transform sets or IKEv2 proposals are allowed for this dynamic crypto map. List multiple transform sets or proposals in order of priority (highest priority first) using the command for IKEv1 transform sets or IKEv2 proposals:

```
crypto dynamic-map dynamic-map-name dynamic-seq-num set ikev1 transform-set
transform-set-name1, [transform-set-name2, …transform-set-name9]

crypto dynamic-map dynamic-map-name dynamic-seq-num set ikev2 ipsec-proposal
proposal-name1
[proposal-name2, … proposal-name11]
```

Dynamic-map-name specifies the name of the crypto map entry that refers to a pre-existing dynamic crypto map. *Dynamic-seq-num* specifies the sequence number that corresponds to the dynamic crypto map entry. The *transform-set-name* is the name of the transform-set being created or modified. The *proposal-name* specifies one or more names of the IPsec proposals for IKEv2.

For example (for IKEv1):

```
crypto dynamic-map dyn 10 set ikev1 transform-set myset1 myset2
```

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

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

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

Dynamic-map-name specifies the name of the crypto map entry that refers to a pre-existing dynamic crypto map. *Dynamic-seq-num* specifies the sequence number that corresponds to the dynamic crypto map entry.

For example:

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

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

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

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

Dynamic-map-name specifies the name of the crypto map entry that refers to a pre-existing dynamic crypto map. *Dynamic-seq-num* specifies the sequence number that corresponds to the dynamic crypto map entry.

For example:

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

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

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

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

Map-name specifies the name of the crypto map set. *Dynamic-map-name* specifies the name of the crypto map entry that refers to a pre-existing dynamic crypto map.

For example:

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

Providing Site-to-Site Redundancy

You can define multiple IKEv1 peers by using crypto maps to provide redundancy. This configuration is useful for site-to-site VPNs. This feature is not supported with IKEv2.

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

Viewing an IPsec Configuration

Table 1-6 lists commands that you can enter in either single or multiple context mode to view information about your IPsec configuration.

Table 1-6 Commands to View IPsec Configuration Information

Command	Purpose
show running-configuration crypto	Displays the entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP.
show running-config crypto ipsec	Displays the complete IPsec configuration.
show running-config crypto isakmp	Displays the complete ISAKMP configuration.
show running-config crypto map	Displays the complete crypto map configuration.
show running-config crypto dynamic-map	Displays the dynamic crypto map configuration.
show all crypto map	Displays all of the configuration parameters, including those with default values.
show crypto ikev2 sa detail	Shows the Suite B algorithm support in the Encryption statistics.

Table 1-6 Commands to View IPsec Configuration Information (continued)

Command	Purpose
show crypto ipsec sa	Shows the Suite B algorithm support and the ESPv3 IPsec output in either single or multiple context mode.
show ipsec stats	Shows information about the IPsec subsystem in either single or multiple context mode. ESPv3 statistics are shown in TFC packets and valid and invalid ICMP errors received.

Clearing Security Associations

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

Table 1-7 lists commands you can enter to clear and reinitialize IPsec SAs in either single or multiple context mode.

Table 1-7 Commands to Clear and Reinitialize IPsec SAs

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

Clearing Crypto Map Configurations

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

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

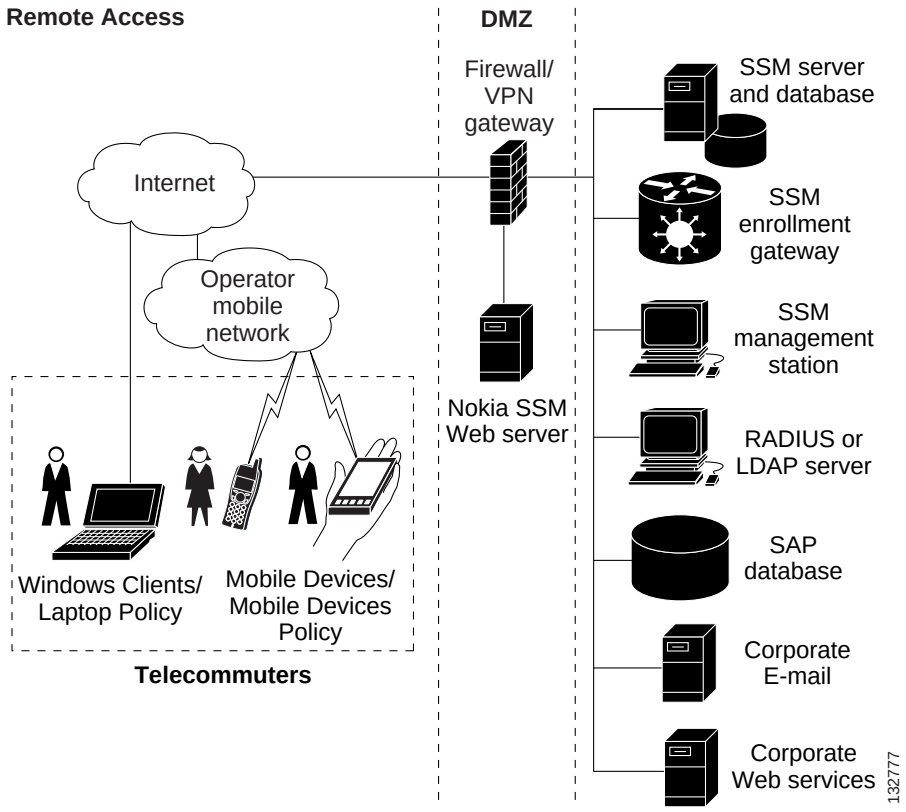
For more information, see the **clear configure crypto** command in the *Cisco ASA Series Command Reference*.

Supporting the Nokia VPN Client

The ASA supports connections from Nokia VPN clients on Nokia 92xx Communicator series phones using the Challenge/Response for Authenticated Cryptographic Keys (CRACK) protocol. CRACK is ideal for mobile IPsec-enabled clients that use legacy authentication techniques instead of digital certificates. It provides mutual authentication when the client uses a legacy-based secret-key authentication technique such as RADIUS and the gateway uses public-key authentication.

The Nokia back-end services must be in place to support both Nokia clients and the CRACK protocol. This requirement includes the Nokia Security Services Manager (NSSM) and Nokia databases as shown in Figure 1-5.

Figure 1-5 Nokia 92xx Communicator Service Requirement



To support the Nokia VPN client, perform the following step on the ASA:

- Enable CRACK authentication using the **crypto isakmp policy** *priority* **authentication** command with the **crack** keyword in global configuration mode. For example:

```
hostname(config)# crypto isakmp policy 2
hostname(config-isakmp-policy)# authentication crack
```

If you are using digital certificates for client authentication, perform the following additional steps:

Step 1 Configure the trustpoint and remove the requirement for a fully qualified domain name. The trustpoint might be NSSM or some other CA. In this example, the trustpoint is named CompanyVPNCA:

```
hostname(config)# crypto ca trustpoint CompanyVPNCA
hostname(config-ca-trustpoint)# fqdn none
```

Step 2 To configure the identity of the ISAKMP peer, perform one of the following steps:

- Use the **crypto isakmp identity** command with the **hostname** keyword. For example:

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

- Use the **crypto isakmp identity** command with the **auto** keyword to configure the identity to be automatically determined from the connection type. For example:

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

Note If you use the **crypto isakmp identity auto** command, you must be sure that the DN attribute order in the client certificate is CN, OU, O, C, St, L.

To learn more about the Nokia services required to support the CRACK protocol on Nokia clients, and to ensure they are installed and configured properly, contact your local Nokia representative.