

# Configuring IPsec

---

This chapter provides information about IP Security Protocol (IPsec), Internet Key Exchange (IKE), IKE Mode Configuration, and Certification Authority (CA) features so that you can successfully implement these features into the PIX Firewall and have Virtual Private Network (VPN) capability. This chapter describes the features' functions, as well as how these features interoperate with one another. You will find the applicable configuration procedures after each component's "about" section.

This chapter includes the following sections:

- Supported Standards
- List of Terms
- Order in Which You Configure Your IPsec
- About IPsec
- Configuring IPsec
- About IKE
- Configuring IKE
- About IKE Mode Configuration (Dynamic IP Address Assignment for Cisco Secure VPN Client)
- Configuring Dynamic IP Addressing Assignment
- About CA
- Configuring CA

The IPsec-related commands are listed and described within Chapter 6, "Command Reference."

See Chapter 5, "Configuration Examples," for additional examples of IPsec configurations.

## Supported Standards

Cisco implements the following standards for the IPSec and IKE features within the PIX Firewall:

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

IPSec is documented in a series of Internet RFCs, all available at <http://www.ietf.org/html.charters/ipsec-charter.html>. The overall IPSec implementation is guided by “Security Architecture for the Internet Protocol,” RFC2401.

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

IPSec as implemented in PIX Firewall supports the following additional standards:

- **AH—Authentication Header.** A security protocol that provides data authentication and optional anti-replay services. AH is embedded in the data to be protected (a full IP datagram).

The AH protocol (RFC2402) allows for the use of various authentication algorithms; PIX Firewall has implemented the mandatory MD5-HMAC (RFC2403) and SHA-HMAC (RFC2404) authentication algorithms. Used in conjunction with ISAKMP, the AH protocol algorithms. In conjunction with ISAKMP, the ESP protocol provides anti-replay services.

- **ESP—Encapsulating Security Payload.** A security protocol that provides data privacy services and optional data authentication, and anti-replay services. ESP encapsulates the data to be protected.

The ESP protocol (RFC2406) allows for the use of various cipher algorithms and (optionally) various authentication algorithms. The PIX Firewall implements the mandatory 56-bit DES-CBC with Explicit IV (RFC2405); as the encryption algorithm, and MD5-HMAC (RFC2403) or SHA-HMAC (RFC2404) as the authentication.

IKE is implemented per the latest version of the “The Internet Key Exchange” Internet Draft (draft-ietf-ipsec-isakmp-oakley-xx.txt).

**ISAKMP—The Internet Security Association and Key Management Protocol.** A protocol framework that defines payload formats, the mechanics of implementing a key exchange protocol, and the negotiation of a security association.

ISAKMP is implemented per the latest version of the “Internet Security Association and Key Management Protocol (ISAKMP)” Internet Draft (draft-ietf-ipsec-isakmp-xx.txt).

**Oakley**—A key exchange protocol that defines how to derive authenticated keying material.

**Skeme**—A key exchange protocol that defines how to derive authenticated keying material, with rapid key refreshment.

The component technologies implemented for use by IKE include:

- **DES—Data Encryption Standard (DES)** is used to encrypt packet data. IKE implements the 56-bit DES-CBC with Explicit IV standard. See “CBC.”
- **Triple DES (3DES)**—A variant of DES, which iterates three times with three separate keys, effectively doubling the strength of DES.

- **CBC**—Cipher Block Chaining (CBC) requires an initialization vector (IV) to start encryption. The IV is explicitly given in the IPSec packet.
- **Diffie-Hellman**—A public-key cryptography protocol which allows two parties to establish a shared secret over an unsecure communications channel. Diffie-Hellman is used within IKE to establish session keys. 768-bit and 1024-bit Diffie-Hellman groups are supported.
- **MD5 (HMAC variant)**—MD5 (Message Digest 5) is a hash algorithm used to authenticate packet data. HMAC is a variant which provides an additional level of hashing.
- **SHA (HMAC variant)**—SHA (Secure Hash Algorithm) is a hash algorithm used to authenticate packet data. HMAC is a variant which provides an additional level of hashing.
- **RSA signatures**—RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA signatures provides non-repudiation.

IKE interoperates with the following standard:

**X.509v3 certificates**—Used with the IKE protocol when authentication requires public keys. Certificate support that allows the IPSec-protected network to scale by providing the equivalent of a digital ID card to each device. When two peers wish to communicate, they exchange digital certificates to prove their identities (thus removing the need to manually exchange public keys with each peer or to manually specify a shared key at each peer). These certificates are obtained from a CA. X.509 is part of the X.500 standard by the ITU.

CA supports the following standards:

- **X.509v3 certificates**
- **Public-Key Cryptography Standard #7 (PKCS #7)**—A standard from RSA Data Security, Inc. used to encrypt and sign certificate enrollment messages.
- **Public-Key Cryptography Standard #10 (PKCS #10)**—A standard syntax from RSA Data Security, Inc. for certificate requests.
- **RSA Keys**—RSA is the public key cryptographic system developed by Ron Rivest, Adi Shamir, and Leonard Adleman. RSA keys come in pairs: one public key and one private key.

## List of Terms

**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. PIX Firewall IPSec provides this service whenever it provides the data authentication service, except in the following cases:

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

**client**—Node or software program (front-end device) that requests services from a server.

**data authentication**—Includes two concepts:

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

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

**data confidentiality**—A security service where the protected data cannot be observed.

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

**gateway**—A device that performs an application layer conversion from one protocol stack to another.

**peer**—In the context of this chapter, a peer refers to a PIX Firewall or other device, such as a Cisco router, that participates in IPSec, IKE, and CA.

**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.

**repudiation**—A quality that prevents a third party from being able to prove that a communication between two other parties ever took place. This is a desirable quality if you do not want your communications to be traceable. Non-repudiation is the opposite quality—a third party can prove that a communication between two other parties took place. Non-repudiation is desirable if you want to be able to trace your communications and prove that they occurred.

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

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

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

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

**transform**—A transform lists a security protocol (AH or ESP) with its corresponding algorithms. For example, one transform is the AH protocol with the MD5-HMAC authentication algorithm; another transform is the ESP protocol with the 56-bit DES encryption algorithm and the SHA-HMAC authentication algorithm.

**tunnel**—In the context of this chapter, a tunnel refers to secure communication path between two peers, such as two PIX Firewall units. It does not refer to using IPSec in tunnel mode.

**Virtual Private Network (VPN)**—Enables IP traffic to travel securely over a public TCP/IP network by encrypting all traffic from one network to another.

## Order in Which You Configure Your IPsec

If you will implement interoperability with a CA, it is recommended that you perform your IPsec configuration in the following order:

- 1 CA (see “Configuring CA”)
- 2 IKE (see “Configuring IKE”)
- 3 (Optional) IKE Mode Configuration—applies only if you are configuring dynamic IP addressing for remote clients (see “Configuring Dynamic IP Addressing Assignment”)
- 4 IPsec (see “Configuring IPsec”)

If you will *not* implement interoperability with a CA, and you will implement IKE, it is recommended that you perform your IPsec configuration in the following order:

- 1 IKE (see “Configuring IKE”)
- 2 (Optional) IKE Mode Configuration—applies only if you are configuring dynamic IP addressing for remote clients (see “Configuring Dynamic IP Addressing Assignment”)
- 3 IPsec (see “Configuring IPsec”)

If you will *not* implement IKE, see “Configuring IPsec.”

---

**Note** Be sure to disable IKE, if you will not implement it.

---

## About IPsec

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

With IPsec, data can be transmitted across a public network without fear of observation, modification, or spoofing. This enables applications such as VPNs, which are categorized by intranets, extranets, and remote dial access. Each VPN type has different security service needs. With VPN, customers, business partners, and remote users, such as telecommuters, can access enterprise computing resources securely. VPNs essentially extend a network’s capability by accommodating the demands of a networked economy for diverse secured connectivity.

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

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

---

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

---

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

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

With IPSec, you define what traffic should be protected between two remote IPSec peers by configuring access lists and applying these access lists to interfaces by way of crypto map sets. Therefore, traffic may be selected on the basis of source and destination address. (Access lists used for IPSec are used only to determine which traffic should be protected by IPSec, not which traffic should be blocked or permitted through the interface. Separate access lists define blocking and permitting at the interface or inbound and outbound from the PIX Firewall.)

---

**Note** Access lists used by IPSec on the PIX Firewall can only contain host selector or greater.

---

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

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

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

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

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

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

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

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

This section includes the following topics, which describe more IPSec conceptual information you will need to know prior to performing your IPSec configuration tasks. The procedures for configuring IPSec are covered in the section “Configuring IPSec.”

- Access Lists' Compatibility with IPSec
- Global Lifetimes for IPSec Security Associations
- Crypto Access Lists
- Transform Sets
- Crypto Map Entries
- Apply Crypto Map Sets to Interface
- Monitor and Maintain IPSec

## Access Lists' Compatibility with IPSec

By default, IPSec and all packets that traverse the PIX Firewall are subjected to blocking as specified by inbound conduit, outbound list or interface access-list. To enable IPSec packets to traverse the PIX Firewall, ensure that you have statements in conduits, outbound lists or interface access-lists that permit the packets. Optionally, **sysopt connection permit-ipsec** can be configured to enable IPSec packets to bypass the conduit, outbound list and interface access-list blocking.

---

**Note** The **sysopt connection permit-ipsec** command enables all packets that are destined to and arriving from an IPSec tunnel to bypass the conduit, outbound list, and interface access-list checks.

---

---

**Note** IPSec packets that are destined to an IPSec tunnel are selected by the crypto map access-list bound to the outgoing interface. IPSec packets that arrive from an IPSec tunnel are authenticated/deciphered by IPSec and subjected to the proxy identity match of the tunnel.

---

## Global Lifetimes for IPSec Security Associations

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

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

There are two lifetimes: a “timed” lifetime and a “traffic-volume” lifetime. A security association expires after the respective lifetime is reached and negotiations will be initiated for 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 new lifetime value will not be applied to currently existing security associations, but will be used in the negotiation of subsequently established security associations. If you wish to use the new values immediately, you can clear all or part of the security association database. Refer to the **clear crypto sa** command for more details.

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

### How These Lifetimes Work

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

The security association (and corresponding keys) will expire according to whichever occurs sooner, either after the seconds timeout or after the kilobytes amount of traffic is passed.

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

If no traffic has passed through the tunnel during the entire life of the security association, a new security association is not negotiated when the lifetime expires.

Instead, a new security association will be negotiated only when IPSec sees another packet that should be protected.

### Crypto Access Lists

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

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

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

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

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

You will then later associate the crypto access lists to the PIX Firewall's outside interface when you configure and apply crypto map sets to this interface.

## Crypto Access List Tips

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

The crypto access list you define will be applied to an interface after you define the corresponding crypto map entry and apply the crypto map set to the interface. Different access lists must be used in different entries of the same crypto map set. However, both inbound and outbound traffic will be evaluated against the same "outbound" IPSec access list. Therefore, the access list's criteria are applied in the forward direction to traffic exiting your PIX Firewall, and the reverse direction to traffic entering your PIX Firewall. In Figure 4-1, IPSec protection is applied to traffic between Host 10.0.0.1 and Host 20.0.0.2 as the data exits PIX Firewall A's outside interface en route to Host 20.0.0.2. For traffic from Host 10.0.0.1 to Host 20.0.0.2, the access list entry on PIX Firewall A is evaluated as follows:

source = host 10.0.0.1

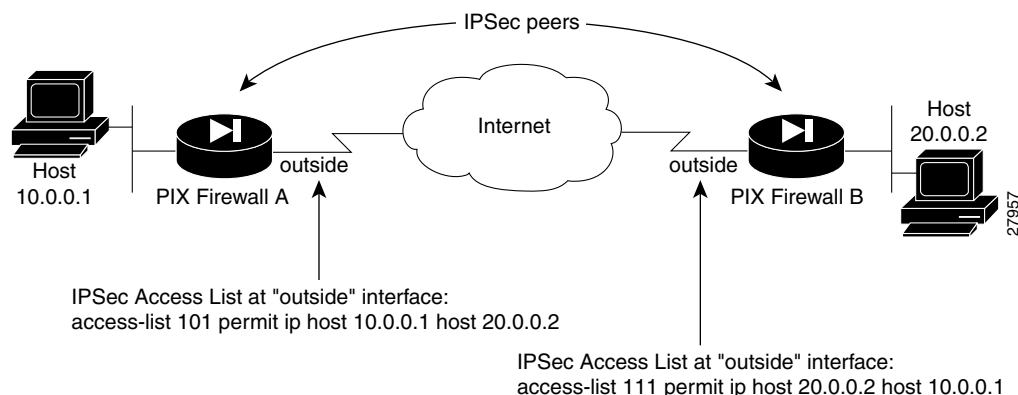
dest = host 20.0.0.2

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

source = host 20.0.0.2

dest = host 10.0.0.1

**Figure 4-1 How Crypto Access Lists Are Applied for Processing IPSec**



Traffic exchanged between hosts 10.0.0.1 and 20.0.0.2  
is protected between PIX Firewall A "outside" and PIX Firewall B "outside"

If you configure multiple statements for a given crypto access list that is used for IPSec, in general the first permit statement that is matched will be the statement used to determine the scope of the IPSec security association. That is, the IPSec security association will be set up to protect traffic that

meets the criteria of the matched statement only. Later, if traffic matches a different permit statement of the crypto access list, a new, separate IPSec security association will be negotiated to protect traffic matching the newly matched access list statement.

---

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

---

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

---

**Note** If you clear or delete the last element from an access list, the crypto map references to the destroyed access list are also removed.

---

---

**Note** If you modify an access list that is currently referenced by one or more crypto map entries, the run-time security association database will need to be re initialized using the **crypto map interface** command. See the **crypto ipsec** command page for information on the **crypto map interface** command.

---

### Mirror Image Crypto Access Lists at each IPSec Peer

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

### any Keyword in Crypto Access Lists

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

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

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

## Transform Sets

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

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

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

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

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

### Selecting Appropriate Transforms

Choosing IPsec transforms combination can be complex. The following tips may help you select transforms that are appropriate for your situation. If you want to:

- provide data confidentiality, include an ESP encryption transform.  
Also consider including an ESP authentication transform or an AH transform to provide authentication services for the transform set.
- ensure data authentication for the outer IP header as well as the data, include an AH transform.
- ensure data authentication (using either ESP or AH) you can choose from the MD5 or SHA (HMAC keyed hash variants) authentication algorithms. The SHA algorithm is generally considered stronger than MD5, but it is slower.

---

**Note** Some transforms may not be supported by the peer.

---

Suggested transform combinations:

- esp-3des and esp-sha-hmac
- esp-des and esp-sha-hmac

## Crypto Map Entries

To create crypto map entries, follow the guidelines described in this section:

- About Crypto Maps
- Load Sharing
- How Many Crypto Maps Should You Create?
- Manual Security Associations (Using Pre-shared Keys)
- IKE Security Associations
- Dynamic Crypto Maps

### About Crypto Maps

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

- Which traffic should be protected by IPSec (per a crypto access list)
- Where IPSec-protected traffic should be sent (who the peer is)
- The local address to be used for the IPSec traffic (See the “Apply Crypto Map Sets to Interface” section for more details.)
- What IPSec security should be applied to this traffic (selecting from a list of one or more transform sets)
- Whether security associations are manually established or are established via IKE
- Other parameters that might be necessary to define an IPSec SA

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

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

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

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

- The crypto map entries must contain compatible crypto access lists (for example, mirror image access lists). In the case where the responding peer is using dynamic crypto maps, the entries in the PIX Firewall crypto access list must be “permitted” by the peer’s crypto access list.

- The crypto map entries must each identify the other peer (unless the responding peer is using dynamic crypto maps).
- The crypto map entries must have at least one transform set in common.

## Load Sharing

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

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

## How Many Crypto Maps Should You Create?

You can apply only one crypto map set to a single interface. The crypto map set can include a combination of IPSec/IKE and IPSec/manual entries.

---

**Note** The PIX Firewall currently only supports IPSec on the outside interface. Although the PIX Firewall currently can simulate the Private Link inside termination with the use of the **sysopt ipsec pl-compatible** command, the termination on the inside interface is not a true termination. The use of the **sysopt ipsec pl-compatible** command allows IPSec packets to bypass the NAT and ASA features, and enables incoming IPSec packets to terminate on the inside interface only after initially terminating on the outside interface. For more information on the **sysopt ipsec pl-compatible** command, see the **sysopt** command page within Chapter 6, “Command Reference.”

---

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

You must create multiple crypto map entries for the PIX Firewall’s outside interface, if any of the following conditions exist:

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

### Manual Security Associations (Using Pre-shared Keys)

The use of manual security associations is a result of a prior arrangement between the users of the PIX Firewall and its peer. There is no negotiation of security associations, so the configuration information in both systems must be the same for traffic to be processed successfully by IPSec.

The PIX Firewall can simultaneously support manual and IKE-established security associations, even within a single crypto map set.

### IKE Security Associations

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

### Dynamic Crypto Maps

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

Dynamic crypto maps are only available for use by IKE.

A dynamic crypto map entry is essentially a crypto map entry without all the parameters configured. It acts as a policy template where the missing parameters are later dynamically configured (as the result of an IPSec negotiation) to match a peer's requirements. This allows peers to exchange IPSec traffic with the PIX Firewall even if the PIX Firewall does not have a crypto map entry specifically configured to meet all the peer's requirements.

---

**Note** Only the transform-set field is required to be configured within each dynamic crypto map entry.

---

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

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

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

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

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

---

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

---

### Dynamic Crypto Map Set

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

If this is configured, the data flow identity proposed by the IPSec peer must fall within a **permit** statement for this crypto access list.

If this is not configured, the PIX Firewall will accept any data flow identity proposed by the peer.

Care must be taken if the **any** keyword is used in the access list, because the access list is used for packet filtering, as well as for negotiation.

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

### Add the Dynamic Crypto Map Set into a Regular (Static) Crypto Map Set

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

## Apply Crypto Map Sets to Interface

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

---

**Note** Although the PIX Firewall currently can simulate the Private Link inside termination with the use of the **sysopt ipsec pl-compatible** command, the termination on the inside interface is not a true termination. For more information on the **sysopt ipsec pl-compatible** command, see the **sysopt** command page within Chapter 6, “Command Reference.”

---



---

**Note** Binding a crypto map to the outside interface will also initialize the run-time data structures, such as the security association database and the security policy database. If the crypto map is modified in any significant manner, reapplying the crypto map to the outside interface will resynchronize the various run-time data structures with the crypto map configuration.

---

## Monitor and Maintain IPSec

Certain configuration changes will only take effect when negotiating subsequent security associations. If you want the new settings to take immediate effect, you must clear the existing security associations so that they will be re-established with the changed configuration. For manually established security associations, you must clear and reinitialize the security associations or the changes will never take effect. If the PIX Firewall 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 PIX Firewall is processing very little other IPSec traffic.

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

Command	Purpose
<b>crypto map</b> <i>map-name</i> <b>interface</b> <i>interface-name</i>	Reinitialize the IPSec run-time security association database and security policy database.
<b>clear crypto sa</b>	Clear IPSec security associations.
or	
<b>clear crypto sa peer</b> <i>ip-address</i>   <i>peer-name</i>	<b>Note</b> Using the <b>clear crypto sa</b> command without parameters will clear out the full security association database, which will clear out active security sessions. You may also specify the <b>peer</b> , <b>map</b> , or <b>entry</b> keywords to clear out only a subset of the security association database. For more information, see the <b>clear crypto sa</b> command within Chapter 6, “Command Reference.”
or	
<b>clear crypto sa map</b> <i>map-name</i>	
or	
<b>clear crypto sa entry</b> <i>destination-address protocol spi</i>	

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

Command	Purpose
<code>show crypto ipsec transform-set</code>	View your transform set configuration.
<code>show crypto map [interface <i>interface-name</i>   tag <i>map-name</i>]</code>	View your crypto map configuration.
<code>show crypto ipsec sa [map <i>map-name</i>   address   identity] [detail]</code>	View information about IPsec security associations.
<code>show crypto dynamic-map [tag <i>map-name</i>]</code>	View information about dynamic crypto maps.
<code>show crypto ipsec security-association lifetime</code>	View global security association lifetime values.

## Configuring IPsec

This section provides procedures to configure IPsec where IPsec security associations will be established via IKE or pre-shared keys (without IKE). See “Configuring IPsec with IKE” to configure IPsec with IKE. Otherwise, see “Configuring Manual IPsec.”

### Configuring IPsec with IKE

The following steps cover minimal IPsec configuration where the IPsec security associations will be established via IKE.

**Step 1** Create an access list to define the traffic to protect:

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

For example:

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

In the above example, the **permit** keyword causes all traffic that matches the specified conditions to be protected by crypto.

**Step 2** Configure a transform set that defines how the traffic will be protected. You can configure multiple transform sets, and then specify one or more of these transform sets in a crypto map entry (Step 3d).

```
crypto ipsec transform-set transform-set-name transform1 [transform2, transform3]
```

For example:

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

```
crypto ipsec transform-set myset2 ah-sha-hmac esp-3des esp-sha-hmac
```

In this example, “myset1” and “myset2” are the names of the transform sets. “myset1” has two transforms defined, while “myset2” has three transforms defined.

**Step 3** Create a crypto map entry by performing the following steps:

(a) Create a crypto map entry in IPsec ISAKMP mode:

```
crypto map map-name seq-num ipsec-isakmp
```

For example:

```
crypto map mymap 10 ipsec-isakmp
```

“mymap” is the name of the crypto map set. The map set’s sequence number is 10, which is used to rank multiple entries within one crypto map set. The lower the sequence number, the higher the priority.

- (b) Assign an access list to a crypto map entry:

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

For example:

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

In the example, access-list 101 is assigned to crypto map “mymap.”

- (c) 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 security association will be set up with the peer having an IP address of 192.168.1.100. Specify multiple peers by repeating this command.

- (d) Specify which transform sets are allowed for this crypto map entry. List multiple transform sets in order of priority (highest priority first). You can specify up to six transform sets.

```
crypto map map-name seq-num set transform-set transform-set-name1
[transform-set-name2, ...transform-set-name6]
```

For example:

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

In this example, when traffic matches access list 101, the security association can use either “myset1” (first priority) or “myset2” (second priority) depending on which transform set matches the peer’s transform set.

- (e) (Optional) Specify security association lifetime for the crypto map entry, if you want the security associations for this entry to be negotiated using different IPsec security association lifetimes other than the global lifetimes.

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

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 2,700 seconds (45 minutes). The traffic volume lifetime is not change.

- (f) (Optional) Specify that IPsec should ask for perfect forward secrecy (PFS) when requesting new security associations for this crypto map entry, or should require PFS in requests received from the peer:

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

For example:

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

This example specifies that PFS should be used whenever a new security association is negotiated for the crypto map “mymap 10.” The 1024-bit Diffie-Hellman prime modulus group will be used when a new security association is negotiated using the Diffie-Hellman exchange.

**Step 4** (Optional) Create a crypto dynamic map entry by performing the following steps:

- (a) Create a dynamic crypto map entry:

```
crypto dynamic-map dynamic-map-name dynamic-seq-num
```

For example:

```
crypto dynamic-map dyn1 10
```

“dyn1” is the name of the dynamic crypto map set. The map set’s sequence number is 10.

- (b) (Optional) Assign an access list to a dynamic crypto map entry, which determines which traffic should be protected and not protected:

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

For example:

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

In the example, access list 101 is assigned to dynamic crypto map “dyn1.”

- (c) (Optional) Specify the peer to which the IPsec-protected traffic can be forwarded. This is *rarely* configured in dynamic crypto map entries because dynamic crypto map entries are often used for unknown peers.

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

For example:

```
crypto dynamic-map dyn1 10 set peer 192.168.1.102
```

The security association will be set up with the peer having an IP address of 192.168.1.102. Specify multiple peers by repeating this command.

- (d) Specify which transform sets are allowed for this dynamic crypto map entry. List multiple transform sets in order of priority (highest priority first).

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

For example:

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

In this example, when traffic matches access list 101, the security association can use either “myset1” (first priority) or “myset2” (second priority) depending on which transform set matches the peer’s transform sets.

- (e) (Optional) Specify security association lifetime for the crypto dynamic map entry, if you want the security associations for this entry to be negotiated using different IPsec security association lifetimes other than the global lifetimes:

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

For example:

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

This example shortens the timed lifetime for dynamic crypto map “dyn1 10” to 2,700 seconds (45 minutes). The time volume lifetime is not changed.

- (f) (Optional) Specify that IPsec should ask for PFS when requesting new security associations for this dynamic crypto map entry, or should demand PFS in requests received from the peer:

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

For example:

```
crypto dynamic-map dyn1 10 set pfs group1
```

- (g) Add the dynamic crypto map set into a static crypto map set.

Be sure to set the crypto map entries 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
```

For example:

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

- Step 5** Apply a crypto map set to the outside interface:

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

For example:

```
crypto map mymap interface outside
```

In this example, the PIX Firewall will evaluate the traffic going through the outside interface against the crypto map “mymap” to determine whether it needs to be protected.

## Configuring Manual IPsec

The following procedure covers minimal IPsec configuration where the security associations will be established via pre-shared keys.

- Step 1** Create an access list to define the traffic to protect:

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

For example:

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

In this example, the keyword **permit** causes all traffic that matches the specified conditions to be protected by crypto.

- Step 2** Configure a transform set that defines how the traffic will be protected.

You can configure only one transform set for manually established security associations. The peer must also have the same transform set specified.

```
crypto ipsec transform-set transform-set-name transform
```

For example:

```
crypto ipsec transform-set myset3 esp-des esp-sha-hmac
```

In the example, “myset3” is the name of the transform set and two transforms have been defined.

**Step 3** Create a crypto map entry in IPsec manual mode:

```
crypto map map-name seq-num ipsec-manual
```

For example:

```
crypto map mymaptwo 30 ipsec-manual
```

**Step 4** Name an IPsec access list. The access list can specify only one permit entry when you are establishing manual security associations.

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

For example:

```
crypto map mymaptwo 30 match address 101
```

**Step 5** Specify the peer to which the IPsec protected traffic can be forwarded. Only one peer can be specified when you are establishing manual security associations.

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

For example:

```
crypto map mymaptwo 30 set peer 192.186.1.103
```

**Step 6** Specify which transform set should be used. This must be the same transform set that is specified in the peer's corresponding crypto map entry.

```
crypto map map-name seq-num set transform-set transform-set-name
```

For example:

```
crypto map mymaptwo 30 set transform-set myset3
```

**Step 7** If the specified transform set includes the AH protocol (authentication via MD5-HMAC or SHA-HMAC), set the AH Security Parameter Index (SPI) and key to apply to inbound protected traffic. If the specified transform set includes only the ESP protocol, skip to Step 9.

```
crypto map map-name seq-num set session-key inbound ah spi hex-key-data
```

For example:

```
crypto map mymaptwo 30 set session-key inbound ah 300  
123456789A123456789A123456789A123456789A
```

This specifies the IPsec session key for AH protocol within crypto map "mymaptwo" to be used with the inbound protected traffic.

**Step 8** Set the AH SPIs and keys to apply to outbound protected traffic:

```
crypto map map-name seq-num set session-key outbound ah spi hex-key-data
```

For example:

```
crypto map mymaptwo 30 set session-key outbound ah 400  
123456789A123456789A123456789A123456789A
```

**Step 9** If the specified transform set includes the ESP protocol, set the ESP SPIs and keys to apply to inbound protected traffic. If the transform set includes an ESP cipher algorithm, specify the cipher keys. If the transform set includes an ESP authenticator algorithm, specify the authenticator keys.

```
crypto map map-name seq-num set session-key inbound esp spi cipher hex-key-data  
[authenticator hex-key-data]
```

For example:

```
crypto map mymaptwo 30 set session-key inbound esp 300 cipher  
123456789012345 authenticator 0000111122223333444455556666777788889999
```

**Step 10** Set the ESP SPIs and keys to apply to inbound protected traffic. If the transform set includes an ESP cipher algorithm, specify the cipher keys. If the transform set includes an ESP authenticator algorithm, specify the authenticator keys.

```
crypto map map-name seq-num set session-key outbound esp spi cipher hex-key-data  
[authenticator hex-key-data]
```

For example:

```
crypto map mymaptwo 30 set session-key outbound esp 300 cipher  
abcdefghijklmnopno authenticator 9999888877776666555544443333222211110000
```

**Step 11** Apply a crypto map set to the outside interface:

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

For example:

```
crypto map mymaptwo interface outside
```

In this example, the PIX Firewall will evaluate the traffic going through the outside interface against the “mymaptwo” crypto map to determine whether it needs to be protected.

## About IKE

IKE is a key management protocol standard that is used in conjunction with the IPsec standard.

IPsec can be configured without IKE, but IKE enhances IPsec by providing additional features, flexibility, and ease of configuration for the IPsec standard.

IKE is a hybrid protocol, which implements the Oakley key exchange and Skeme key exchange inside the ISAKMP framework. (ISAKMP, Oakley, and Skeme are security protocols implemented by IKE.) IKE automatically negotiates IPsec security associations and enables IPsec secure communications without manual preconfiguration.

Specifically, IKE provides these benefits:

- Eliminates the need to manually specify all the IPsec security parameters in the crypto maps at both peers.
- Allows you to specify a lifetime for the IPsec security association.
- Allows encryption keys to change during IPsec sessions.
- Allows IPsec to provide anti-replay services.
- Permits CA support for a manageable, scalable IPsec implementation.
- Allows dynamic authentication of peers.

This section includes the following topics, which provide background information applicable to the tasks of configuring IKE. The IKE configuration steps are provided in the section “Configuring IKE.”

- IKE Policies
- IKE Pre-shared Keys

## IKE Policies

You must create IKE policies at each peer. An IKE policy defines a combination of security parameters to be used during IKE negotiation.

To create an IKE policy, follow the guidelines in these topics:

- Why Do You Need to Create These Policies?
- What Parameters Do You Define in a Policy?
- How Do IKE Peers Agree upon a Matching Policy?
- Which Value Should You Select for Each Parameter?
- Creating Policies
- Additional Configuration Required for IKE Policies

### Why Do You Need to Create These Policies?

IKE negotiations must be protected, so each IKE negotiation begins by each peer agreeing on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations.

After the two peers agree upon a policy, the security parameters of the policy are identified by a security association established at each peer, and these security associations apply to all subsequent IKE traffic during the negotiation.

You can create multiple, prioritized policies at each peer to ensure that at least one policy will match a remote peer's policy.

### What Parameters Do You Define in a Policy?

There are five parameters to define in each IKE policy. These parameters apply to the IKE negotiations when the IKE security association is established. Table 4-1 provides the five IKE policy parameters and their accepted values.

**Table 4-1 IKE Policy Parameters**

Parameter	Accepted Values	Keyword	Default Value
encryption algorithm	56-bit DES-CBC	des	56-bit DES-CBC
	168-bit Triple DES	3des	
hash algorithm	SHA-1 (HMAC variant)	sha	SHA-1
	MD5 (HMAC variant)	md5	
authentication method	RSA signatures	rsa-sig	RSA signatures
	pre-shared keys	pre-share	
Diffie-Hellman group identifier	768-bit Diffie-Hellman or	1	768-bit Diffie-Hellman
	1024-bit Diffie-Hellman	2	
security association's lifetime	can specify any number of seconds	–	86,400 seconds (one day)

### How Do IKE Peers Agree upon a Matching Policy?

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 made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer's policy specifies a lifetime less than or equal to the lifetime in the policy being compared. (If the lifetimes are not identical, the shorter lifetime—from the remote peer's policy—will be used.)

If no acceptable match is found, IKE refuses negotiation and IPSec will not be established.

If a match is found, IKE will complete negotiation, and IPSec security associations will be created.

---

**Note** Depending on which authentication method is specified in a policy, additional configuration might be required (as described in the section “Additional Configuration Required for IKE Policies”). If a peer's policy does not have the required companion configuration, the peer will not submit the policy when attempting to find a matching policy with the remote peer.

---

### Which Value Should You Select for Each Parameter?

You can select certain values for each parameter, per the IKE standard. But why chose one value over another?

If you are interoperating with a peer that supports only one of the values for a parameter, your choice is limited to the other peer's supported value. Aside from this, there is often a trade-off between security and performance, and many of these parameter values represent such a trade-off. You should evaluate the level of your network's security risks and your tolerance for these risks. Then the following tips might help you select which value to specify for each parameter.

- The encryption algorithm has two options: 56-bit DES and 168-bit triple DES.
- The hash algorithm has two options: SHA-1 and MD5.

MD5 has a smaller digest and is considered to be slightly faster than SHA-1. There has been a demonstrated successful (but extremely difficult) attack against MD5; however, the HMAC variant used by IKE prevents this attack.

- The authentication method has two options: RSA signatures and pre-shared keys.

RSA signatures provide non-repudiation for the IKE negotiation (you can prove to a third party after the fact that you had an IKE negotiation with a specific peer).

RSA signatures requires use of a CA. Using a CA can dramatically improve the manageability and scalability of your IPSec network.

Pre-shared keys are clumsy to use if your secured network is large, and do not scale well with a growing network. However, they do not require use of a CA, as do RSA signatures, and might be easier to set up in a small network with fewer than 10 nodes.

- The Diffie-Hellman group identifier has two options: 768-bit or 1024-bit Diffie-Hellman.

1024-bit Diffie-Hellman is more secure than the 768-bit Diffie-Hellman, but requires more CPU time to execute.

- The SA's lifetime can be set to any value.

As a general rule, the shorter the lifetime (up to a point), the more secure your IKE negotiations will be. However, with longer lifetimes, future IPSec security associations can be set up more quickly.

## Creating Policies

You can create multiple IKE policies, each with a different combination of parameter values. For each policy that you create, you assign a unique priority (1 through 65,534, with 1 being the highest priority).

You can configure multiple policies on each peer—but at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer. (The lifetime parameter does not necessarily have to be the same; see details in the section “How Do IKE Peers Agree upon a Matching Policy?”)

If you do not configure any policies, your PIX Firewall will use the default policy, which is always set to the lowest priority, and which contains each parameter’s default value.

If you do not specify a value for a parameter, the default value is assigned.

## Additional Configuration Required for IKE Policies

Depending on which authentication method you specify in your IKE policies, you need to do certain additional configuration before IKE and IPSec can successfully use the IKE policies.

Each authentication method requires additional companion configuration as follows:

- RSA signatures method:

If you specify RSA signatures as the authentication method in a policy, you must configure the peers to obtain certificates from a CA. (And, of course, the CA must be properly configured to issue the certificates.) Configure this certificate support as described in the section “Configuring CA.”

The certificates are used by each peer to securely exchange public keys. (RSA signatures require that each peer has the remote peer’s public signature key.) When both peers have valid certificates, they will automatically exchange public keys with each other as part of any IKE negotiation in which RSA signatures are used.

- Pre-shared keys authentication method:

If you specify pre-shared keys as the authentication method in a policy, you must configure these pre-shared keys as described in “Configuring Pre-shared (Authentication) Keys Manually.”

## IKE Pre-shared Keys

To configure pre-shared keys, perform one or both of the following tasks at each peer that uses pre-shared keys in an IKE policy. See “Configuring Pre-shared (Authentication) Keys Manually” for the procedure that tells you how to configure pre-shared keys.

- You will need to set each peer’s ISAKMP identity, if its identity is to be set to hostname. Each peer’s identity is set either to its host name or its IP address. When two peers use IKE to establish IPSec security associations, each peer sends its identity to its peer.

By default, a peer’s identity is its IP address. If appropriate, you could change the identity to be the peer’s host name instead. As a general rule, set all peers’ identities the same way—either all peers should use their IP addresses or all peers should use their host names. If some peers use

their host names and some peers use their IP addresses to identify themselves to one another, IKE negotiations could fail if a peer's identity is not recognized and a DNS lookup is unable to resolve the identity.

- Specify the shared keys at each peer. A given pre-shared key is shared between two peers. At a given peer you could specify the same key to share with multiple peers; however, a more secure approach is to specify different keys to share between different pairs of peers.

## Configuring IKE

This section provides the procedures to enable and configure IKE. It also provides a procedure to disable IKE, if you choose not use it with your IPSec implementation.

The following topics are included in this section:

- Enabling and Configuring IKE
- Configuring Pre-shared (Authentication) Keys Manually
- Disabling IKE

## Enabling and Configuring IKE

To enable and configure IKE, perform the following steps:

- Step 1** Enable IKE on the outside interface:

```
isakmp enable interface-name
```

For example:

```
isakmp enable outside
```

- Step 2** Configure the IKE policies by performing the following steps:

---

**Note** If you enter a default value for a given policy parameter, it will not be written in the configuration. If you do not specify a value for a given policy parameter, the default value is assigned.

---

- (a) Identify the policy to create. Each policy is uniquely identified by the priority number you assign.

```
isakmp policy priority
```

For example:

```
isakmp policy 20 ...
```

- (b) Specify the encryption algorithm:

```
isakmp policy priority encryption des|3des
```

For example:

```
isakmp policy 20 encryption des
```

- (c) Specify the hash algorithm:

```
isakmp policy priority hash md5|sha
```

For example:

```
isakmp policy 20 hash md5
```

- (d) Specify the authentication method:

```
isakmp policy priority authentication pre-share|rsa-sig
```

For example:

```
isakmp policy 20 authentication rsa-sig
```

---

**Note** If you specify the authentication method of pre-shared keys, you are required to manually configure these keys. See “Configuring Pre-shared (Authentication) Keys Manually.”

---

- (e) Specify the Diffie-Hellman group identifier:

```
isakmp policy priority group 1|2
```

For example:

```
isakmp policy 20 group 2
```

- (f) Specify the security association’s lifetime:

```
isakmp policy priority lifetime seconds
```

For example:

```
isakmp policy 20 lifetime 5000
```

The following example shows two policies with policy 20 as the highest priority, policy 30 as the next priority, and the existing default policy as the lowest priority:

```
isakmp policy 20 encryption des
isakmp policy 20 hash md5
isakmp policy 20 authentication rsa-sig
isakmp policy 20 group 2
isakmp policy 20 lifetime 5000

isakmp policy 30 authentication pre-share
isakmp policy 30 lifetime 10000
```

In this example, the encryption des of policy 20 would not appear in the written configuration because this is the default for the encryption algorithm parameter.

**Step 3** (Optional) View all existing IKE policies:

```
show isakmp policy
```

The following is an example of the output after the policies 20 and 30 in the previous example were configured:

```
Protection suite priority 20
  encryption algorithm:  DES - Data Encryption Standard (56 bit keys)
  hash algorithm:        Message Digest 5
  authentication method: Rivest-Shamir-Adleman Signature
  Diffie-Hellman group:  #2 (1024 bit)
  lifetime:              5000 seconds, no volume limit
Protection suite priority 30
  encryption algorithm:  DES - Data Encryption Standard (56 bit keys)
  hash algorithm:        Secure Hash Standard
  authentication method: Pre-Shared Key
  Diffie-Hellman group:  #1 (768 bit)
  lifetime:              10000 seconds, no volume limit
Default protection suite
  encryption algorithm:  DES - Data Encryption Standard (56 bit keys)
  hash algorithm:        Secure Hash Standard
  authentication method: Rivest-Shamir-Adleman Signature
  Diffie-Hellman group:  #1 (768 bit)
  lifetime:              86400 seconds, no volume limit
```

---

**Note** Although the output shows “no volume limit” for the lifetimes, you can currently only configure a time lifetime (such as 86,400 seconds) with IKE; volume limit lifetimes are not currently configurable.

---

## Configuring Pre-shared (Authentication) Keys Manually

If you selected the authentication method of pre-shared keys within Step 2d in the section “Enabling and Configuring IKE,” manually configure these keys between the PIX Firewall and its peer(s). To configure these pre-shared keys on the PIX Firewall, perform the following step:

Specify the pre-shared key that the PIX Firewall and its peer will use for authentication and the peer’s address:

```
isakmp key keystring address peer-address [netmask mask]
```

For example:

```
isakmp key 1234567890 address 192.168.1.100
```

The pre-shared key is 1234567890, and the peer’s address is 192.168.1.100

---

**Note** Netmask allows you to configure a single key to be shared among multiple peers. You would use the netmask of 0.0.0.0. However, Cisco strongly recommends using a unique key for each peer.

---



---

**Note** The pre-shared key must be configured at both the PIX Firewall and its peer, otherwise the policy cannot be used.

---

## Disabling IKE

IKE is enabled in the PIX Firewall by default. If you do not want IKE to be used with your IPSec implementation, disable it.

If you disable IKE, you will have to make these concessions at the peers:

- You must manually specify all the IPSec security associations in the crypto maps at all peers.
- IPSec security associations will never time out for a given IPSec session.
- The encryption keys never change during IPSec sessions between peers.
- Anti-replay services will not be available between the peers.
- CA support cannot be used.

To disable IKE, use the following command:

```
no crypto isakmp enable interface-name
```

For example:

```
no crypto isakmp enable outside
```

## About IKE Mode Configuration (Dynamic IP Address Assignment for Cisco Secure VPN Client)

The IKE Mode Configuration allows a gateway (in this case the PIX Firewall) to download an IP address (and other network level configuration) to the client (peer) as part of an IKE negotiation. Using this exchange, the PIX Firewall gives IP addresses to the IKE client to be used as an “inner” IP address encapsulated under IPSec. This provides a known IP address for the client, which can be matched against the IPSec policy.

The following topics are covered in this section to provide background information applicable to the tasks of configuring the IKE Mode Configuration. The procedure to configure the IKE Mode Configuration is provided in the section “Configuring Dynamic IP Addressing Assignment.”

- Benefits
- Types
- Configuration Tasks

### Benefits

To implement IPSec VPNs between remote access clients with dynamic IP addresses and a corporate gateway, you have to dynamically administer scalable IPSec policy on the gateway once each client is authenticated. With IKE Mode Configuration, the gateway can set up scalable policy for a very large set of clients irrespective of the IP addresses of those clients.

### Types

There are two types of IKE Mode Configuration for VPN:

- Gateway initiation—Gateway initiates the configuration mode with the client. Once the client responds, the IKE modifies the sender's identity, the message is processed, and the client receives a response.
- Client initiation—Client initiates the configuration mode with the gateway. The gateway responds with an IP address it has allocated for the client.

### Configuration Tasks

The following are the three steps to perform when configuring IKE Mode Configuration on your PIX Firewall. The next section, "Configuring Dynamic IP Addressing Assignment," gives you the procedure.

- Define the pool of IP addresses. Existing local address pools are used to define a set of addresses. Use the **ip local pool** command to define a local address pool. See the **ip** command page within Chapter 6, "Command Reference," for more information about this command.
- Reference the pool of IP addresses in the IKE configuration. Use the **isakmp client configuration address-pool local** command to configure the IP address local pool you defined to reference IKE. See the **isakmp** command page within Chapter 6, "Command Reference," for more information about this command.
- Define which crypto maps should attempt to configure clients, and whether the PIX Firewall or the client initiates the IKE Mode Configuration. Use the **crypto map client-configuration address** command to configure IKE Mode Configuration. See the **crypto map** command page within Chapter 6, "Command Reference," for more information about this command.

## Configuring Dynamic IP Addressing Assignment

To configure IKE Mode Configuration on your PIX Firewall, perform the following steps after configuring IKE.

**Step 1** Define the pool of IP addresses:

```
ip local pool pool-name start-address-[end-address]
```

For example:

```
ip local pool ire 171.72.1.1-171.72.1.254
```

**Step 2** Reference the defined pool of IP addresses in the IKE configuration:

```
isakmp client configuration address-pool local pool-name [interface-name]
```

For example:

```
isakmp client configuration address-pool local ire outside
```

**Step 3** Define which crypto maps should attempt to configure clients:

```
crypto map map-name client configuration address initiate|respond
```

For example:

```
crypto map mymap client configuration address initiate
```

## Examples

The following partial configuration shows a PIX Firewall that has been configured to both set IP addresses to clients and respond to IP address requests from clients whose packets arrive on the outside interface using dynamic crypto map without explicitly specifying the peer. The IKE Mode Configuration commands are in bold.

---

**Note** The section “VPN Client Access with AAA and Pre-shared Keys” in Chapter 5, “Configuration Examples,” provides another example of configuring IKE Mode Configuration.

---

```

! define the ip address pool
ip local pool ire 171.72.1.1-171.72.1.254
! tie the pool with ike
crypto isakmp client configuration address-pool local ire outside
!
access-list 103 permit ip host 172.21.230.34 172.21.1.0 255.255.255.0
!
crypto ipsec transform-set pc esp-des esp-md5-hmac
!
crypto dynamic-map dyn 10 set transform-set pc
crypto dynamic-map dyn 10 match address 103
! enable address assignment in crypto map
crypto map dyn client configuration address initiate
crypto map dyn client configuration address respond
!
crypto map dyn 10 ipsec-isakmp dynamic dyn
crypto map dyn interface outside

```

## About CA

CAs are responsible for managing certificate requests and issuing certificates to participating IPsec network peers. These services provide centralized key management for the participating peers.

CAs simplify the administration of IPsec network devices (peers). You can use a CA with a network containing multiple IPsec-compliant devices, such as with the Cisco Secure PIX Firewalls and Cisco routers.

Digital signatures, enabled by public key cryptography, provide a means to digitally authenticate devices and individual users. In public key cryptography, such as the RSA encryption system, each user has a key-pair containing both a public and a private key. The keys act as complements, and anything encrypted with one of the keys can be decrypted with the other. In simple terms, a signature is formed when data is encrypted with a user’s private key. The receiver verifies the signature by decrypting the message with the sender’s public key. The fact that the message could be decrypted using the sender’s public key indicates that the holder of the private key, the sender, must have created the message. This process relies on the receiver having a copy of the sender’s public key and knowing with a high degree of certainty that it really does belong to the sender, and not to someone pretending to be the sender.

Digital certificates provide this link. A digital certificate contains information to identify a user or device, such as the name, serial number, company, department or IP address. It also contains a copy of the entity’s public key. The certificate is itself signed by a Certification Authority, a third party that is explicitly trusted by the receiver to validate identities and to create digital certificates.

In order to validate the CA’s signature, the receiver must first know the CA’s public key. Normally this is handled out-of-band or through an operation done at installation. For instance, most web browsers are configured with the public keys of several CAs by default. The IKE, a key component of IPSec, can use digital signatures to scaleably authenticate peer devices before setting up security associations.

Without digital signatures, one must either manually exchange public keys or secrets between each pair of peers that use IPSec to protect communications between them. Without certificates, every new peer added to the network requires a configuration change on every other peer it securely communicates with. However, by using digital certificates, each peer is enrolled with a CA. When two peers wish to communicate, they exchange certificates and digitally sign data to authenticate each other. When a new peer is added to the network, one simply enrolls that peer with a CA, and none of the other peers need modification. When the new peer attempts an IPSec connection, certificates are automatically exchanged and the peer can be authenticated.

The following topics are covered in this section to provide background information applicable to the the tasks of configuring for interoperability with a CA. The CA configuration steps are provided in “CA Configuration.”

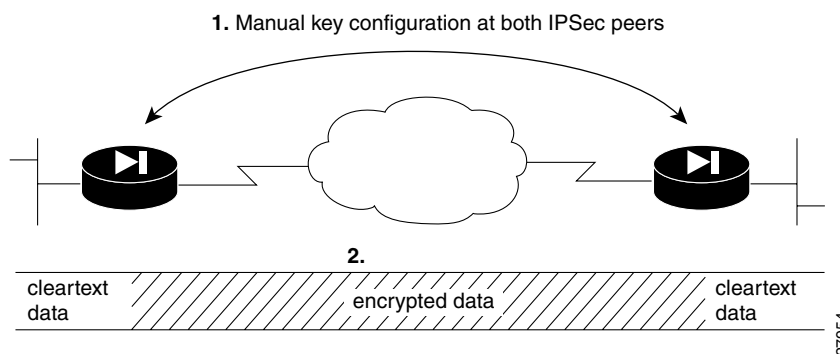
- IPSec without CAs
- IPSec with CAs
- How CA Certificates Are Used by IPSec Peers
- Registration Authorities

## IPSec without CAs

Without a CA, if you want to enable IPSec services (such as encryption) between two peers, you must first ensure that each peer has the other’s key (such as an RSA public key or a shared key). This requires that you manually perform one of the following at each peer:

- Enter the other peer’s RSA public key, or
- Specify a shared key to be used between the two peers.

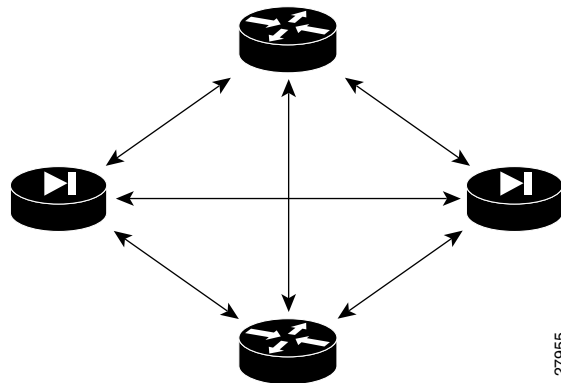
**Figure 4-2 Without a CA: Key Configuration between two IPSec Peers**



In Figure 4-2, number 1 illustrates how each PIX Firewall uses the other's key to authenticate the identity of the other PIX Firewall; this authentication always occurs whenever IPSec traffic is exchanged between two IPSec peers. Number 2 illustrates encrypted data within an IPSec tunnel between two IPSec peers.

If you have multiple Cisco peers in a mesh topology, and wish to exchange IPSec traffic passing between all of those peers, you must first configure shared keys or RSA public keys between all of those peers:

**Figure 4-3 Without a CA: Six 2-Part Key Configurations Required for Four IPSec Peers**



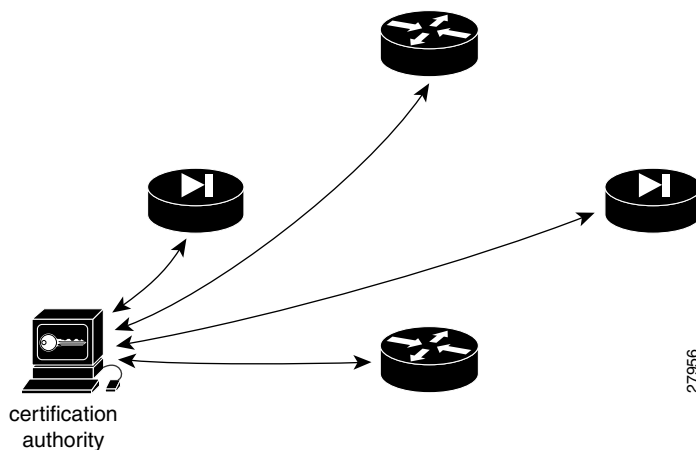
Every time a new peer is added to the IPSec network, you must configure keys between the new peer and each of the existing peers. (In Figure 4-3, four additional 2-part key configurations would be required to add a single IPSec peer to the network.)

Consequently, the more peers there are that require IPSec services, the more involved the key administration becomes. This approach does not scale well for larger, more complex encrypting networks.

## IPSec with CAs

With a CA, you do not need to configure keys between all of the encrypting devices (peers). Instead, you individually enroll each participating device with the CA, requesting a certificate for the device. When this has been accomplished, each participating device can dynamically authenticate all of the other participating peers. This is illustrated in Figure 4-4.

**Figure 4-4 With a CA: Each IPSec Device Individually Makes Request of the CA**



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

## How CA Certificates Are Used by IPSec Peers

When two IPSec peers 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.

Without a CA, a device authenticates itself to the remote peer using either RSA-encrypted nonces or pre-shared keys. PIX Firewall currently does *not* support RSA-encrypted nonces. Both methods require that keys must have been previously configured between the two peers.

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

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

CAs can also revoke certificates for peers that will no longer participate in IPSec. Revoked certificates are not recognized as valid by other peers. Revoked certificates are listed in a Certificate Revocation List (CRL), which each peer may check before accepting another peer's certificate.

## Registration Authorities

Some CAs have a Registration Authority (RA) as part of their implementation. An RA is essentially a server that acts as a proxy for the CA so that CA functions can continue when the CA is offline.

Some of the configuration tasks described in this section differ slightly depending on whether your CA supports an RA or not.

## Configuring CA

To enable your PIX Firewall to interoperate with a CA and obtain your PIX Firewall certificate(s), complete the following steps:

---

**Note** Be sure that the PIX Firewall clock is set to the current time, month, day, and year before configuring CA. Otherwise, the CA may reject or allow certificates based on an incorrect timestamp. Cisco's PKI protocol uses the clock to make sure that a CRL is not expired.

---

**Note** You need to have a CA available to your network before you configure CA. The CA must support Cisco's PKI protocol, the certificate enrollment protocol.

---

**Step 1** Configure the PIX Firewall host name:

```
hostname newname
```

For example:

```
hostname mypixfirewall
```

In this example, "mypixfirewall" is the name of a unique host in the domain.

**Step 2** Configure the PIX Firewall domain name:

```
domain-name name
```

For example:

```
domain-name example.com
```

**Step 3** Generate the PIX Firewall RSA key pair(s):

```
ca generate rsa key key_modulus_size
```

For example:

```
ca generate rsa key 512
```

In this example, one general purpose RSA key pair is to be generated. The other option is to generate two special-purpose keys. The selected size of the key modulus is 512.

**Step 4** (Optional) View your RSA key pair(s):

```
show ca mypubkey rsa
```

The following is an example of the display after the **show ca mypubkey rsa** command is used:

```
% Key pair was generated at: 15:34:55 Aug 05 1999

Key name: mypixfirewall.example.com
Usage: General Purpose Key
Key Data:
 305c300d 06092a86 4886f70d 01010105 00034b00 30480241 00c31f4a ad32f60d
 6e7ed9a2 32883ca9 319a4b30 e7470888 87732e83 c909fb17 fb5cae70 3de738cf
 6e2fd12c 5b3ffa98 8c5adc59 1ec84d78 90bdb53f 2218cfe7 3f020301 0001
```

**Step 5** Declare a CA:

```
ca identity ca_nickname ca_ipaddress [:ca_script_location] [ldap_ip address]
```

For example:

```
ca identity myca.example.com 205.139.94.230
```

In this example, 205.139.94.230 is the IP address of the CA, which is onsiteipsec.verisign.com. The CA name is myca.example.com.

---

**Note** The CA may require a particular name for you to use, such as its domain name.

---

---

**Note** When using VeriSign as your CA, VeriSign assigns the CA name you are to use in your CA configuration.

---

**Step 6** Configure the parameters of communication between the PIX Firewall and the CA:

```
ca configure ca_nickname ca|ra retry_period retry_count [crloptional]
```

For example:

```
ca configure myca.example.com ca 1 20 crloptional
```

If the PIX Firewall does not receive a certificate from the CA within 1 minute (default) of sending a certificate request, it will resend the certificate request. The firewall will continue sending a certificate request every 1 minute until a certificate is received or until 20 requests have been sent. With the keyword **crloptional** included within the command statement, other peer's certificates can still be accepted by your firewall even if the CRL is not accessible to your firewall.

**Step 7** Authenticate the CA by obtaining its public key and its certificate.

```
ca authenticate ca_nickname [fingerprint]
```

For example:

```
ca authenticate myca.example.com 0123 4567 89AAB CDEF 0123
```

The fingerprint (0123 4567 89AAB CDEF 0123 in the example) is optional and is used to authenticate the CA's public key within its certificate. The PIX Firewall will discard the CA certificate if the fingerprint that you included in the command statement is not equal to the fingerprint within the CA's certificate.

You also have the option to manually authenticate the public key by simply comparing the two fingerprints after you receive the CA's certificate rather than entering it within the command statement.

---

**Note** Depending on the CA you are using, you may need to ask your local CA administrator for this fingerprint.

---

- Step 8** Request signed certificates from your CA for all of your PIX Firewall's RSA key pairs. Before entering this command, contact your CA administrator because he or she will have to authenticate your PIX Firewall manually before granting its certificate(s).

```
ca enroll ca_nickname challenge_password [serial] [ipaddress]
```

For example:

```
ca enroll myca.example.com mypassword1234567
```

The keyword `mypassword1234567` in the example is a password, which is not saved with the configuration.

---

**Note** The password is required in the event your certificate needs to be revoked, so it is crucial that you remember this password. Note it and store it in a safe place.

---

The **ca enroll** command requests as many certificates as there are RSA key pairs. You will only need to perform this command once, even if you have special usage RSA key pairs.

---

**Note** If your PIX Firewall reboots after you issued the **ca enroll** command but before you received the certificate(s), you must reissue the command and notify the CA administrator.

---

- Step 9** Verify that the enrollment process was successful:

```
show ca certificate
```

Here is an example of the output of the **show ca certificate** command including a PIX Firewall general purpose certificate and the RA and CA public-key certificates:

```
Subject Name
  Name: mypixfirewall.example.com
IP Address: 192.150.50.110
  Status: Available
  Certificate Serial Number: 36f97573
  Key Usage: General Purpose

RA Signature Certificate
  Status: Available
  Certificate Serial Number: 36f972f4
  Key Usage: Signature

CA Certificate
  Status: Available
  Certificate Serial Number: 36f972e5
  Key Usage: Not Set

RA KeyEncipher Certificate
  Status: Available
  Certificate Serial Number: 36f972f3
  Key Usage: Encryption
```

- Step 10** Save your configuration.

```
ca save all
write memory
```

