CHAPTER 27

Configuring Internet Key Exchange (IKE) Features Using the IPSec VPN SPA

This chapter provides information about configuring Internet Key Exchange (IKE) related features using the IPSec VPN SPA on the Cisco 7600 series router. It includes the following sections:

- Overview of IKE, page 27-2
- Configuring Advanced Encryption Standard in an IKE Policy Map, page 27-3
- Configuring ISAKMP Keyrings and Peer Filtering, page 27-4
- Configuring Certificate to ISAKMP Profile Mapping, page 27-5
- Configuring an Encrypted Preshared Key, page 27-9
- Configuring IKE Aggressive Mode Initiation, page 27-11
- Configuring Call Admission Control for IKE, page 27-13
- Configuring Dead Peer Detection, page 27-15
- Configuring IPSec NAT Transparency, page 27-17
- Configuration Examples, page 27-19

For detailed information on Internet Key Exchange (IKE), refer to the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference*.

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* and *Cisco IOS Configuration Fundamentals Command Reference* publications.

For more information about the commands used in this chapter, see first Chapter 37, “SIP, SSC, and SPA Commands,” and then the *Cisco 7600 Series Cisco IOS Command Reference* publication. Also refer to the related Cisco IOS software command reference and master index publications. For more information about accessing these publications, see the “Related Documentation” section on page -xl.

Tip To ensure a successful configuration of your VPN using the IPSec VPN SPA, read all of the configuration summaries and guidelines before you perform any configuration tasks.

Overview of 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 automatically negotiates IPSec security associations (SAs) and enables IPSec secure communications without costly manual preconfiguration. Specifically, IKE provides the following 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 SA.
- Allows encryption keys to change during IPSec sessions.
- Allows IPSec to provide anti-replay services.
- Permits certification authority (CA) support for a manageable, scalable IPSec implementation.
- Allows dynamic authentication of peers.

Because IKE negotiations must be protected, each IKE negotiation begins by agreement of both peers on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated. You must create an IKE policy at each peer participating in the IKE exchange.

If you do not configure any IKE policies, your router will use the default policy, which is always set to the lowest priority and which contains the default value of each parameter.

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

You can configure multiple, prioritized policies on each peer—each with a different combination of parameter values. However, 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. For each policy that you create, you assign a unique priority (1 through 10,000, with 1 being the highest priority).

Configuring Advanced Encryption Standard in an IKE Policy Map

The Advanced Encryption Standard (AES) is a privacy transform for IPSec and Internet Key Exchange (IKE) that has been developed to replace the Data Encryption Standard (DES). AES is designed to be more secure than DES. AES offers a larger key size, while ensuring that the only known approach to decrypt a message is for an intruder to try every possible key. AES has a variable key length—the algorithm can specify a 128-bit key (the default), a 192-bit key, or a 256-bit key.

To configure the AES encryption algorithm within an IKE policy map, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto isakmp policy** *priority*	Defines an ISAKMP policy and enters ISAKMP policy configuration mode. • *priority*—Identifies the IKE policy and assigns a priority to the policy. Use an integer from 1 to 10000, with 1 being the highest priority and 10000 the lowest.
Step 2	Router(config-isakmp)# **encryption** {**aes** | **aes 192** | **aes 256**}	Specifies the encryption algorithm within an IKE policy. • **aes**—Specifies 128-bit AES as the encryption algorithm. • **aes 192**—Specifies 192-bit AES as the encryption algorithm. • **aes 256**—Specifies 256-bit AES as the encryption algorithm.
Step 3	**...** Router(config-isakmp) # **exit**	Specify any other policy values appropriate to your configuration, and then exit ISAKMP policy configuration mode. For details on configuring an ISAKMP policy, see the *Cisco IOS Security Configuration Guide*.

Verifying the AES IKE Policy

To verify the configuration of the AES IKE policy, enter the **show crypto isakmp policy** command:

```
Router# show crypto isakmp policy

Protection suite of priority 1
encryption algorithm:  AES - Advanced Encryption Standard (256 bit keys).
hash algorithm:        Secure Hash Standard
authentication method: Pre-Shared Key
Diffie-Hellman group:  #1 (768 bit)
lifetime: 3600 seconds, no volume limit
```

For an AES configuration example, see the "Advanced Encryption Standard Configuration Example" section on page 27-19.

Configuring ISAKMP Keyrings and Peer Filtering

The SafeNet IPSec VPN Client Support feature allows you to limit the scope of an Internet Security Association and Key Management Protocol (ISAKMP) profile or ISAKMP keyring configuration to a local termination address or interface. The benefit of this feature is that different customers can use the same peer identities and ISAKMP keys by using different local termination addresses.

ISAKMP Keyrings and Peer Filtering Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring ISAKMP keyrings and peer filtering:

- The local address option works only for the primary address of an interface.
- If an IP address is provided, the administrator must ensure that the connection of the peer terminates to the address that is provided.
- If the IP address does not exist on the device, or if the interface does not have an IP address, the ISAKMP profile or ISAKMP keyring will be effectively disabled.

Limiting an ISAKMP Profile to a Local Termination Address or Interface

To configure an ISAKMP profile and limit it to a local termination address or interface, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto isakmp profile** *profile-name*	Defines an ISAKMP profile and enters ISAKMP profile configuration mode. • *profile-name*—Name of the ISAKMP profile.
Step 2	Router(conf-isa-profile)# **keyring** *keyring-name*	(Optional) Configures a keyring with an ISAKMP profile. • *keyring-name*—Name of the crypto keyring. **Note** A keyring is not needed inside an ISAKMP profile for local termination to work. Local termination works even if Rivest, Shamir, and Adelman (RSA) certificates are used.
Step 3	Router(conf-isa-profile)# **match identity address** *address*	Matches an identity from a peer in an ISAKMP profile. • *address*—IP address of the remote peer.
Step 4	Router(conf-isa-profile)# **local-address** {*interface-name* | *ip-address* [*vrf-tag*]}	Limits the scope of an ISAKMP profile or an ISAKMP keyring configuration to a local termination address or interface. • *interface-name*—Name of the local interface. • *ip-address*—Local termination address. • *vrf-tag*—(Optional) Scope of the IP address will be limited to the VRF.

Limiting a Keyring to a Local Termination Address or Interface

To configure an ISAKMP keyring and limit its scope to a local termination address or interface, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **keyring** *keyring-name*	Defines a crypto keyring to be used during IKE authentication and enters keyring configuration mode. • *keyring-name*—Name of the crypto keyring.
Step 2	Router(conf-keyring)# **local-address** {*interface-name* | *ip-address* [*vrf-tag*]}	Limits the scope of an ISAKMP profile or an ISAKMP keyring configuration to a local termination address or interface. • *interface-name*—Name of the local interface. • *ip-address*—Local termination address. • *vrf-tag*—(Optional) Scope of the IP address will be limited to the VRF.
Step 3	Router(conf-keyring)**# pre-shared-key address** *address*	Defines a preshared key to be used for IKE authentication. • *address*—IP address.

For complete configuration information for ISAKMP keyrings and peer filtering, refer to this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_14/gt_scse.htm

For ISAKMP keyrings and peer filtering configuration examples, see the "ISAKMP Keyrings and Peer Filtering Configuration Examples" section on page 27-20.

Configuring Certificate to ISAKMP Profile Mapping

The Certificate to ISAKMP Profile Mapping feature enables you to assign an Internet Security Association and Key Management Protocol (ISAKMP) profile to a peer on the basis of the contents of arbitrary fields in the certificate. In addition, this feature allows you to assign a group name to those peers that are assigned an ISAKMP profile.

Certificate to ISAKMP Profile Mapping Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring Certificate to ISAKMP Profile Mapping:

- This feature will not be applicable if you use Rivest, Shamir, and Adelman- (RSA-) signature or RSA-encryption authentication without certificate exchange. ISAKMP peers must be configured to do RSA-signature or RSA-encryption authentication using certificates.

Mapping the Certificate to the ISAKMP Profile

To map the certificate to the ISAKMP profile, perform the following steps beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto isakmp profile** *profile-name*	Defines an ISAKMP profile and enters ISAKMP profile configuration mode • *profile-name*—Name of the user profile.
Step 2	Router(config-isa-prof)# **match certificate** *certificate-map*	Accepts the name of a certificate map. • *certificate-map*—Name of the certificate map.

Verifying the Certificate to ISAKMP Profile Mapping Configuration

To verify that the subject name of the certificate map has been properly configured, enter the **show crypto pki certificates** and the **debug crypto isakmp** commands.

The **show crypto pki certificates** command displays all current IKE security associations (SAs) at a peer. The **debug crypto isakmp** command displays messages about IKE events.

The following examples show that a certificate has been mapped to an ISAKMP profile. The examples include the configurations for the responder and initiator, the **show crypto pki certificates** command output verifying that the subject name of the certificate map has been configured, and **the debug crypto isakmp** command output showing that the certificate has gone through certificate map matching and been matched to the ISAKMP profile.

Responder Configuration:

```
crypto pki certificate map cert_map 10
! The above line is the certificate map definition.
 subject-name co ou = green
! The above line shows that the subject name must have "ou = green."
!
crypto isakmp profile certpro
! The above line shows that this is the ISAKMP profile that will match if the certificate
of the peer matches cert_map (shown on third line below).
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
   initiate mode aggressive
```

Initiator Configuration:

```
crypto ca trustpoint LaBcA
 enrollment url http://10.76.82.20:80/cgi-bin/openscep
 subject-name ou=green,c=IN
! The above line ensures that the subject name "ou = green" is set.
 revocation-check none
```

show crypto pki certificates Command Output for the Initiator:

```
Router# show crypto pki certificates
Certificate
  Status: Available
  Certificate Serial Number: 21
  Certificate Usage: General Purpose
  Issuer:
    cn=blue-lab CA
    o=CISCO
    c=IN
  Subject:
    Name: Router1.cisco.com
    c=IN
    ou=green
! The above line is a double check that "ou = green" has been set as the subject name.
    hostname=Router1.cisco.com
  Validity Date:
    start date: 14:34:30 UTC Mar 31 2004
    end   date: 14:34:30 UTC Apr 1 2009
    renew date: 00:00:00 UTC Jan 1 1970
  Associated Trustpoints: LaBcA
```

debug crypto isakmp Command Output for the Responder:

```
Router# debug crypto isakmp
6d23h: ISAKMP (0:268435460): received packet from 192.0.0.2 dport 500 sport 500 Global (R) 
MM_KEY_EXCH
6d23h: ISAKMP: Main Mode packet contents (flags 1, len 892):
6d23h:           ID payload
6d23h:             FQDN <Router1.cisco.com> port 500 protocol 17
6d23h:           CERT payload
6d23h:           SIG payload
6d23h:           KEEPALIVE payload
6d23h:           NOTIFY payload
6d23h: ISAKMP:(0:4:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
6d23h: ISAKMP:(0:4:HW:2):Old State = IKE_R_MM4  New State = IKE_R_MM5
6d23h: ISAKMP:(0:4:HW:2): processing ID payload. message ID = 0
6d23h: ISAKMP (0:268435460): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : Router1.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
6d23h: ISAKMP:(0:4:HW:2):: peer matches *none* of the profiles
6d23h: ISAKMP:(0:4:HW:2): processing CERT payload. message ID = 0
6d23h: ISAKMP:(0:4:HW:2): processing a CT_X509_SIGNATURE cert
6d23h: ISAKMP:(0:4:HW:2): peer's pubkey isn't cached
6d23h: ISAKMP:(0:4:HW:2): OU = green
6d23h: ISAKMP:(0:4:HW:2): certificate map matches certpro profile
! The above line shows that the certificate has gone through certificate map matching and 
that it matches the "certpro" profile.
6d23h: ISAKMP:(0:4:HW:2): Trying to re-validate CERT using new profile
6d23h: ISAKMP:(0:4:HW:2): Creating CERT validation list: 2315, LaBcA,
6d23h: ISAKMP:(0:4:HW:2): CERT validity confirmed.
```

Assigning the Group Name to the Peer

To associate a group name with an ISAKMP profile that will be assigned to a peer, use the **client configuration group** command beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto isakmp profile** *profile-name*	Defines an ISAKMP profile and enters ISAKMP profile configuration mode • *profile-name*—Name of the user profile.
Step 2	Router (conf-isa-prof)# **client configuration group** *group-name*	Accepts the name of a group that will be assigned to a peer when the peer is assigned this crypto ISAKMP profile. • ***group-name***—Name of the group to be associated with the peer.

Verifying the Group Name to Peer Assignation Configuration

To verify that a group has been assigned to a peer, enter the **debug crypto isakmp** command.

The **debug crypto isakmp** command displays messages about IKE events.

The following **debug crypto isakmp** output shows that the peer has been matched to the ISAKMP profile named "certpro" and that it has been assigned a group named "new_group."

Initiator Configuration:

```
crypto isakmp profile certpro
   ca trust-point 2315
   ca trust-point LaBcA
   match certificate cert_map
   client configuration group new_group
! The statement on the above line will assign the group "new_group" to any peer that
matches the ISAKMP profile "certpro."
   initiate mode aggressive
```

debug crypto isakmp Command Output for the Responder:

```
Router# debug crypto isakmp
6d23h: ISAKMP (0:268435461): received packet from 192.0.0.2 dport 500 sport 500 Global (R)
MM_KEY_EXCH
6d23h: ISAKMP: Main Mode packet contents (flags 1, len 892):
6d23h:           ID payload
6d23h:             FQDN <Router1.cisco.com> port 500 protocol 17
6d23h:           CERT payload
6d23h:           SIG payload
6d23h:           KEEPALIVE payload
6d23h:           NOTIFY payload
6d23h: ISAKMP:(0:5:HW:2):Input = IKE_MESG_FROM_PEER, IKE_MM_EXCH
6d23h: ISAKMP:(0:5:HW:2):Old State = IKE_R_MM4  New State = IKE_R_MM5
6d23h: ISAKMP:(0:5:HW:2): processing ID payload. message ID = 0
6d23h: ISAKMP (0:268435461): ID payload
        next-payload : 6
        type         : 2
        FQDN name    : Router1.cisco.com
        protocol     : 17
        port         : 500
        length       : 28
6d23h: ISAKMP:(0:5:HW:2):: peer matches *none* of the profiles
6d23h: ISAKMP:(0:5:HW:2): processing CERT payload. message ID = 0
```

```
6d23h: ISAKMP:(0:5:HW:2): processing a CT_X509_SIGNATURE cert
6d23h: ISAKMP:(0:5:HW:2): peer's pubkey isn't cached
6d23h: ISAKMP:(0:5:HW:2): OU = green
6d23h: ISAKMP:(0:5:HW:2): certificate map matches certpro profile
6d23h: ISAKMP:(0:5:HW:2): Trying to re-validate CERT using new profile
6d23h: ISAKMP:(0:5:HW:2): Creating CERT validation list: 2315, LaBcA,
6d23h: ISAKMP:(0:5:HW:2): CERT validity confirmed.
6d23h: ISAKMP:(0:5:HW:2):Profile has no keyring, aborting key search
6d23h: ISAKMP:(0:5:HW:2): Profile certpro assigned peer the group named new_group
```

For complete configuration information for certificate to ISAKMP profile mapping, refer to this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_8/gt_isakp.htm

For certificate to ISAKMP profile mapping configuration examples, see the "Certificate to ISAKMP Profile Mapping Configuration Examples" section on page 27-21.

Configuring an Encrypted Preshared Key

The Encrypted Preshared Key feature allows you to securely store plain text passwords in type 6 (encrypted) format in NVRAM.

Encrypted Preshared Key Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring an encrypted preshared key:

- Old ROM monitors (ROMMONs) and boot images cannot recognize the new type 6 passwords. Therefore, errors are expected if you boot from an old ROMMON.
- If the password (master key) is changed, or reencrypted, using the **key config-key password-encryption** command, the list registry passes the old key and the new key to the application modules that are using type 6 encryption.
- If the master key that was configured using the **key config-key password-encryption** command is deleted from the system, a warning is printed (and a confirm prompt is issued) that states that all type 6 passwords will become useless. As a security measure, after the passwords have been encrypted, they will never be decrypted in the Cisco IOS software. However, passwords can be reencrypted.

Caution If the password configured using the **key config-key password-encryption** command is lost, it cannot be recovered. The password should be stored in a safe location.

- If you later unconfigure password encryption using the **no password encryption ae**s command, all existing type 6 passwords are left unchanged, and as long as the password (master key) that was configured using the **key config-key password-encryption** command exists, the type 6 passwords will be decrypted as and when required by the application.
- Because no one can "read" the password (configured using the **key config-key password-encryption** command), there is no way that the password can be retrieved from the router. Existing management stations cannot "know" what it is unless the stations are enhanced to include this key somewhere, in which case the password needs to be stored securely within the management system. If configurations are stored using TFTP, the configurations are not standalone, meaning that they cannot be loaded onto a router. Before or after the configurations are loaded onto

a router, the password must be manually added (using the **key config-key password-encryption** command). The password can be manually added to the stored configuration but is not recommended because adding the password manually allows anyone to decrypt all passwords in that configuration.

- If you enter or cut and paste cipher text that does not match the master key, or if there is no master key, the cipher text is accepted or saved, but the following alert message is printed:

```
ciphertext>[for username bar>] is incompatible with the configured master key
```

- If a new master key is configured, all the plain keys are encrypted and made type 6 keys. The existing type 6 keys are not encrypted. The existing type 6 keys are left as is.
- If the old master key is lost or unknown, you have the option of deleting the master key using the **no key config-key password-encryption** command. Deleting the master key using the **no key config-key password-encryption** command causes the existing encrypted passwords to remain encrypted in the router configuration. The passwords will not be decrypted.

Configuring an Encrypted Preshared Key

To configure an encrypted preshared key, perform the following steps beginning global configuration mode:

	Command	Purpose

Step 1	Router(config)# **key config-key password-encryption**	Stores a type 6 encryption key in private NVRAM.
		Note the following:
		• If you are entering the key interactively (using the **Enter** key) and an encrypted key already exists, you will be prompted for the following:
		`Old key, New key, and Confirm key`
		• If you are entering the key interactively but an encryption key is not present, you will be prompted for the following:
		`New key and Confirm key`
		• If you are removing a password that is already encrypted, you will see the following prompt:
		`WARNING: All type 6 encrypted keys will become unusable. Continue with master key deletion? [yes/no]:`

Step 2	Router(config)# **password-encryption aes**	Enables the encrypted preshared key.

Verifying the Encrypted Preshared Key Configuration

To verify that a new master key has been configured and that the keys have been encrypted with the new master key, enter the **password logging** command. The following is an example of its output:

```
Router# password logging

Router (config)# key config-key password-encrypt

New key:
Confirm key:
Router (config)#
01:40:57: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master keypas
```

```
Router (config)# key config-key password-encrypt
Old key:
New key:
Confirm key:
Router (config)#
01:42:11: TYPE6_PASS: Master key change heralded, re-encrypting the keys
with the new master key
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
01:42:11: TYPE6_PASS: Mac verification successful
```

For complete configuration information for the Encrypted Preshared Key feature, refer to this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_2/gt_epsk.htm

For an encrypted preshared key configuration example, see the "Encrypted Preshared Key Configuration Example" section on page 27-21.

Configuring IKE Aggressive Mode Initiation

The IKE: Initiate Aggressive Mode feature allows you to configure Internet Key Exchange (IKE) preshared keys as RADIUS tunnel attributes for IP Security (IPSec) peers.

Although IKE preshared keys are easy to deploy, they do not scale well with an increasing number of users and are therefore prone to security threats. Instead of keeping your preshared keys on the hub router, this feature allows you to scale your preshared keys by storing and retrieving them from an authentication, authorization, and accounting (AAA) server. The preshared keys are stored in the AAA server as Internet Engineering Task Force (IETF) RADIUS tunnel attributes and are retrieved when a user tries to "speak" to the hub router. The hub router retrieves the preshared key from the AAA server and the spokes (the users) initiate aggressive mode to the hub by using the preshared key that is specified in the Internet Security Association and Key Management Protocol (ISAKMP) peer policy as a RADIUS tunnel attribute.

IKE Aggressive Mode Initiation Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring IKE Aggressive Mode Initiation:

- IKE Aggressive Mode Initiation is not intended to be used with a dynamic crypto map that uses Tunnel Endpoint Discovery (TED) to initiate tunnel setup. TED is useful in configuring a full-mesh setup, which requires an AAA server at each site to store the preshared keys for the peers; this configuration is not practical for use with this feature.
- Only the following ID types can be used in this feature:
  - ID_IPV4 (IPV4 address)
  - ID_FQDN (fully qualified domain name, for example "foo.cisco.com")
  - ID_USER_FQDN (e-mail address)
- Before configuring IKE Aggressive Mode Initiation, you must perform the following tasks:
  - Configure AAA
  - Configure an IPSec transform
  - Configure a static crypto map
  - Configure an ISAKMP policy

– Configure a dynamic crypto map

For information on completing these tasks, refer to the *Cisco IOS Security Configuration Guide*.

To configure IKE Aggressive Mode Initiation, you must configure the Tunnel-Client-Endpoint and Tunnel-Password attributes within the ISAKMP peer configuration. To do this, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	Router(config)# **crypto map** *map-name* **isakmp authorization list** *list-name*	Enables IKE querying of AAA for tunnel attributes in aggressive mode. • *map-name*—Name you assign to the crypto map set. • *list-name*—Character string used to name the list of authorization methods activated when a user logs in. The list name must match the list name defined during AAA configuration.
Step 2	Router(config)# **crypto isakmp peer** {**ip-address** *ip-address* | **fqdn** *fqdn*}	Enables an IPSec peer for IKE querying of AAA for tunnel attributes in aggressive mode and enters ISAKMP policy configuration mode. • *ip-address*—IP address of the peer router. • *fqdn*—Fully-qualified domain name of the peer router.
Step 3	Router(config-isakmp)# **set aggressive-mode client-endpoint** *client-endpoint*	Specifies the Tunnel-Client-Endpoint attribute within an ISAKMP peer configuration. • *client-endpoint*—One of the following ID types of the initiator end of the tunnel: – ID_IPV4 (IPV4 address) – ID_FQDN (fully qualified domain name, for example “foo.cisco.com”) – ID_USER_FQDN (e-mail address) **Note** The ID type is translated to the corresponding ID type in Internet Key Exchange (IKE).
Step 4	Router(config-isakmp)# **set aggressive-mode password** *password*	Specifies the Tunnel-Password attribute within an ISAKMP peer configuration. • *password*—Password that is used to authenticate the peer to a remote server. The tunnel password is used as the Internet Key Exchange (IKE) preshared key.

Verifying the IKE Aggressive Mode Initiation Configuration

To verify that the Tunnel-Client-Endpoint and Tunnel-Password attributes have been configured within the ISAKMP peer policy, use the **show running-config** global configuration command.

For complete configuration information for IKE Aggressive Mode Initiation, refer to this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t8/ft_ikeag.htm

For IKE Aggressive Mode Initiation configuration examples, see the "IKE Aggressive Mode Initiation Configuration Examples" section on page 27-22.

Configuring Call Admission Control for IKE

Call Admission Control for IKE allows you to limit the number of simultaneous IKE security associations (SAs) that a router can establish.

There are two ways to limit the number of IKE SAs that a router can establish to or from another router:

- Configure an absolute IKE SA limit by entering the **crypto call admission limit** command. When an IKE SA limit is defined, the router drops new IKE SA requests when this value has been reached as follows. When there is a new SA request from a peer router, IKE determines if the number of active IKE SAs plus the number of SAs being negotiated meets or exceeds the configured SA limit. If the number is greater than or equal to the limit, the new SA request is rejected and a syslog is generated. This log contains the source destination IP address of the SA request.
- Configure a system resource limit by entering the **call admission limit** command. When a system resource limit is defined, the router drops new IKE SA requests when the specified percentage of system resources is being used as follows. CAC polls a global resource monitor so that IKE knows when the router is running short of CPU cycles or memory buffers. You can configure a resource limit, from 1 to 100, that represents a percentage of system resources. When that percentage of the system resources is being used, IKE drops (will not accept new) SA requests. For example, if you specify a resource limit of 90 percent, IKE stops accepting SA requests when 90 percent of the system resources is being used.

CAC is applied only to new SAs (that is, when an SA does not already exist between the peers). Every effort is made to preserve existing SAs. Only new SA requests will ever be denied due to a lack of system resources or because the configured IKE SA limit has been reached.

Configuring the IKE Security Association Limit

To configure an IKE Security Association limit, perform the following steps beginning in global configuration mode. When an IKE SA limit is defined, the router drops new IKE SA requests when this value has been reached:

	Command	Purpose
Step 1	Router(config)# **crypto call admission limit ike sa** *number*	Specifies the maximum number of IKE SAs that the router can establish before IKE begins rejecting new SA requests. • *number*—Number of active IKE SAs allowed on the router. The value must be greater than 1.
Step 2	Router (config)# **exit**	Returns to privileged EXEC mode.

Configuring a System Resource Limit

To configure a system resource limit, perform the following steps beginning in global configuration mode. When an IKE SA limit is defined, the router drops new IKE SA requests he router drops new IKE SA requests when the specified percentage of system resources is being used.

	Command	Purpose
Step 1	Router(config)# **call admission limit** *percent*	Instructs IKE to stop accepting new SA requests (that is, calls for CAC) when the specified percentage of system resources is being used. • *percent*—Percentage of the system resources that, when used, causes IKE to stop accepting new SA requests. Valid values are 1 to 100.
Step 2	Router (config)# **exit**	Returns to privileged EXEC mode.

Clearing Call Admission Statistics

To clear the Call Admission Control counters that track the number of accepted and rejected Internet Key Exchange (IKE) requests, use the **clear crypto call admission statistics** command in global configuration mode:

Router(config)# **clear crypto call admission statistics**

Verifying the Call Admission Control for IKE Configuration

To verify that Call Admission Control has been configured, enter the **show call admission statistics** and the **show crypto call admission statistics** commands.

The **show call admission statistics** command monitors the global CAC configuration parameters and the behavior of CAC.

```
Router# show call admission statistics
Total Call admission charges: 0, limit 25
Total calls rejected 12, accepted 51
Load metric: charge 0, unscaled 0
```

The **show crypto call admission statistics** command monitors crypto CAC statistics.

```
Router# show crypto call admission statistics
-----------------------------------------------------------
               Crypto Call Admission Control Statistics
-----------------------------------------------------------
System Resource Limit: 0    Max IKE SAs 0
Total IKE SA Count:    0    active:     0   negotiating: 0
Incoming IKE Requests: 0    accepted:   0   rejected:    0
Outgoing IKE Requests: 0    accepted:   0   rejected:    0
Rejected IKE Requests: 0    rsrc low:   0   SA limit:    0
```

For more complete configuration information for Call Admission Control for IKE, refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_8/gtcallik.htm

For Call Admission Control for IKE configuration examples, see the "Call Admission Control for IKE Configuration Examples" section on page 27-23.

Configuring Dead Peer Detection

Dead Peer Detection (DPD), defined in RFC 3706, is a mechanism used to detect dead IPSec peers. IPSec is a peer-to-peer type of technology. It is possible that IP connectivity may be lost between peers due to routing problems, peer reloading, or some other situation. This lost connectivity can result in black holes where traffic is lost. DPD, based on a traffic-detection method, is one possible mechanism to remedy this situation.

DPD supports two options: on-demand or periodic. The on-demand approach is the default. With on-demand DPD, messages are sent on the basis of traffic patterns. For example, if a router has to send outbound traffic and the liveliness of the peer is questionable, the router sends a DPD message to query the status of the peer. If a router has no traffic to send, it never sends a DPD message. If a peer is dead, and the router never has any traffic to send to the peer, the router will not find out until the IKE or IPSec security association (SA) has to be rekeyed (the liveliness of the peer is unimportant if the router is not trying to communicate with the peer). On the other hand, if the router has traffic to send to the peer, and the peer does not respond, the router will initiate a DPD message to determine the state of the peer.

With the periodic option, you can configure your router so that DPD messages are "forced" at regular intervals. This forced approach results in earlier detection of dead peers. For example, if a router has no traffic to send, a DPD message is still sent at regular intervals, and if a peer is dead, the router does not have to wait until the IKE SA times out to find out.

DPD is configured using the **crypto isakmp keepalive** command. DPD and Cisco IOS keepalives function on the basis of a timer. If the timer is set for 10 seconds, the router will send a "hello" message every 10 seconds (unless, of course, the router receives a "hello" message from the peer). The benefit of IOS keepalives and periodic DPD is earlier detection of dead peers. However, IOS keepalives and periodic DPD rely on periodic messages that have to be sent with considerable frequency. The result of sending frequent messages is that the communicating peers must encrypt and decrypt more packets.

DPD and IOS keepalive features can be used in conjunction with multiple peers in the crypto map to allow for stateless failover. DPD allows the router to detect a dead IKE peer, and when the router detects the dead state, the router deletes the IPSec and IKE SAs to the peer. If you configure multiple peers, the router will switch over to the next listed peer for a stateless failover.

Dead Peer Detection Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring DPD:

- If you do not configure the **periodic** option, the router defaults to the **on-demand** approach.
- Before configuring periodic DPD, you should ensure that your IKE peer supports DPD. Implementations that support DPD include the Cisco VPN 3000 concentrator, Cisco PIX Firewall, Cisco VPN Client, and Cisco IOS software in all modes of operation—site-to-site, Easy VPN remote, and Easy VPN server.
- Using periodic DPD potentially allows the router to detect an unresponsive IKE peer with better response time when compared to on-demand DPD. However, use of periodic DPD incurs extra overhead. When communicating to large numbers of IKE peers, you should consider using on-demand DPD instead.
- When the **crypto isakmp keepalive** command is configured, the Cisco IOS software negotiates the use of Cisco IOS keepalives or DPD, depending on which protocol the peer supports.

Configuring a Dead Peer Detection Message

To allow the router to send DPD messages to the peer, enter the **crypto isakmp keepalive** command in global configuration mode as follows:

```
Router(config)# crypto isakmp keepalive seconds [retries] [periodic |on-demand]
```

In this command:

- *seconds* specifies the number of seconds between DPD messages; the range is from 10 to 3600 seconds.
- *retries* (Optional) specifies the number of seconds between DPD retries if the DPD message fails; the range is from 2 to 60 seconds. If unspecified, the default is 2 seconds.
- **periodic** (Optional) specifies the that DPD messages are sent at regular intervals.
- **on-demand** (Optional) specifies DPD retries are sent on demand. This is the default behavior.

Note Because the **on-demand** option is the default, the **on-demand** keyword does not appear in configuration output.

Configuring Dead Peer Detection and Cisco IOS Keepalives with Multiple Peers in a Crypto Map

To configure DPD and IOS keepalives to be used in conjunction with the crypto map to allow for stateless failover, perform the following steps beginning in global configuration mode. This configuration will cause a router to cycle through the peer list when it detects that the first peer is dead:

	Command	Purpose
Step 1	Router(config)# **crypto map** *map-name seq-num* **ipsec-isakmp**	Enters crypto map configuration modes and creates or modifies a crypto map entry. • *map-name*—Name that identifies the map set. • *seq-num*—Sequence number assigned to the crypto map entry. • **ipsec-isakmp**—Indicates that IKE will be used to establish the IPSec SAs for protecting the traffic specified by this crypto map entry.
Step 2	Router(config-crypto-map)# **set peer** {*host-name* [**dynamic**] |*ip-address*}	Specifies an IPSec peer in a crypto map entry. • *hostname*—IPSec peer host name. • **dynamic**—(Optional) Indicates that the host name of the IPSec peer will be resolved via a domain name server (DNS) lookup right before the router establishes the IPSec tunnel. • *ip-address*—IPSec peer IP address. You can specify multiple peers by repeating this command.

	Command	Purpose
Step 3	Router(config-crypto-map)# **set transform-set** *transform-set-name*	Specifies which transform sets can be used with the crypto map entry. • *transform-set-name*—Name of the transform set. You can specify more than one transform set name by repeating this command.
Step 4	Router(config-crypto-map)# **match address** [*access-list-id* | *name*]	Specifies an extended access list for a crypto map entry. • *access-list-id*—(Optional) Identifies the extended access list by its name or number. This value should match the access-list-number or name argument of the extended access list being matched. • *name*—(Optional) Identifies the named encryption access list. This name should match the name argument of the named encryption access list being matched.

Verifying the Dead Peer Detection Configuration

To verify that DPD is enabled, enter the **debug crypto isakmp** command:

```
Router(config)# debug crypto isakmp

*Mar 25 15:17:14.131: ISAKMP:(0:1:HW:2):IKE_DPD is enabled, initializing timers
```

For more complete configuration information for Cisco IOS Dead Peer Detection (DPD), refer to the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123newft/123t/123t_7/gtdpmo.htm

For Dead Peer Detection configuration examples, see the "Dead Peer Detection Configuration Examples" section on page 27-23.

Configuring IPSec NAT Transparency

The IPSec NAT Transparency feature introduces support for IP Security (IPSec) traffic to travel through Network Address Translation (NAT) or Point Address Translation (PAT) points in the network by addressing many known incompatibilities between NAT and IPSec.

Before the introduction of this feature, a standard IPSec virtual private network (VPN) tunnel would not work if there were one or more NAT or PAT points in the delivery path of the IPSec packet. This feature makes NAT IPSec-aware, thereby allowing remote access users to build IPSec tunnels to home gateways.

IPSec NAT Transparency Configuration Guidelines and Restrictions

Follow these guidelines and restrictions when configuring IPSec NAT transparency:

- For non-DMVPN configurations, NAT transparency is supported in both tunnel and transport mode.
- For DMVPN configurations, NAT transparency is only supported in transport mode.

Configuring NAT Transparency

NAT Transparency is a feature that is auto-detected by the IPSec VPN SPA. There are no configuration steps. If both VPN devices are NAT transparency capable, NAT Transparency is auto-detected and auto-negotiated.

Disabling NAT Transparency

You may wish to disable NAT Transparency if you already know that your network uses IPSec-awareness NAT (SPI-matching scheme). To disable NAT Transparency, use the following command from global configuration mode:

```
Router(config)# no crypto ipsec nat-transparency udp-encapsulation
```

Configuring NAT Keepalives

To configure your router to send NAT keepalive packets, enter the crypto isakmp nat keepalive command in global configuration mode:

```
Router(config)# crypto isakmp nat keepalive seconds
```

In this command, *seconds* specifies the number of seconds between keepalive packets; range is between 5 to 3,600 seconds.

Verifying the NAT Keepalives Configuration

To verify the NAT keepalives configuration, enter the show crypto ipsec sa command:

```
Router# show crypto ipsec sa

interface:GigabitEthernet5/0/1
    Crypto map tag:testtag, local addr. 10.2.80.161

    local ident (addr/mask/prot/port):(10.2.80.161/255.255.255.255/0/0)
    remote ident (addr/mask/prot/port):(100.0.0.1/255.255.255.255/0/0)
    current_peer:100.0.0.1:4500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps:109, #pkts encrypt:109, #pkts digest 109
    #pkts decaps:109, #pkts decrypt:109, #pkts verify 109
    #pkts compressed:0, #pkts decompressed:0
    #pkts not compressed:0, #pkts compr. failed:0, #pkts decompress failed:0
    #send errors 90, #recv errors 0

    local crypto endpt.:10.2.80.161, remote crypto endpt.:100.0.0.1:4500
    path mtu 1500, media mtu 1500
    current outbound spi:23945537

    inbound esp sas:
    spi:0xF423E273(4095992435)
    transform:esp-des esp-sha-hmac ,
    in use settings ={Tunnel UDP-Encaps, }
    slot:0, conn id:200, flow_id:1, crypto map:testtag
    sa timing:remaining key lifetime (k/sec):(4607996/2546)
    IV size:8 bytes
    replay detection support:Y

    inbound ah sas:

    inbound pcp sas:
```

```
    outbound esp sas:
    spi:0x23945537(596923703)
    transform:esp-des esp-sha-hmac ,
    in use settings ={Tunnel UDP-Encaps, }
    slot:0, conn id:201, flow_id:2, crypto map:testtag
    sa timing:remaining key lifetime (k/sec):(4607998/2519)
    IV size:8 bytes
    replay detection support:Y

    outbound ah sas:

    outbound pcp sas:
```

For complete configuration information for Cisco IOS IPSec NAT Transparency support, refer to this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122newft/122t/122t13/ftipsnat.htm

For a NAT keepalives configuration example, see the "IPSec NAT Transparency Configuration Example" section on page 27-24.

Configuration Examples

This section provide examples of the following configurations:

- Advanced Encryption Standard Configuration Example, page 27-19
- ISAKMP Keyrings and Peer Filtering Configuration Examples, page 27-20
- Certificate to ISAKMP Profile Mapping Configuration Examples, page 27-21
- Encrypted Preshared Key Configuration Example, page 27-21
- IKE Aggressive Mode Initiation Configuration Examples, page 27-22
- Call Admission Control for IKE Configuration Examples, page 27-23
- Dead Peer Detection Configuration Examples, page 27-23
- IPSec NAT Transparency Configuration Example, page 27-24

Advanced Encryption Standard Configuration Example

The following example is sample output from the **show running-config** command. In this example, the Advanced Encryption Standard (AES) 256-bit key is enabled.

```
Router# show running-config

Current configuration : 1665 bytes
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname "Router1"
ip subnet-zero
no ip domain lookup
ip audit notify log
ip audit po max-events 100
crypto isakmp policy 10
encryption aes 256
authentication pre-share
lifetime 180
```

```
crypto isakmp key cisco123 address 10.0.110.1
crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac
mode transport
crypto map aesmap 10 ipsec-isakmp
set peer 10.0.110.1
set transform-set aesset
Router(config)# crypto ipsec ipv4 deny-policy clear
IRouter(conf-isa-profile)# keyring keyring1
Router(conf-isa-profile)# match identity address 10.0.0.0 255.0.0.0
Router(conf-isa-profile)# local-address serial2/0
```

ISAKMP Keyrings and Peer Filtering Configuration Examples

The following examples show how to limit the scope of an Internet Security Association and Key Management Protocol (ISAKMP) profile or ISAKMP keyring configuration to a local termination address or interface:

- ISAKMP Profile Bound to a Local Interface Configuration Example, page 27-20
- ISAKMP Keyring Bound to a Local Interface Configuration Example, page 27-20
- ISAKMP Keyring Bound to a Local IP Address Configuration Example, page 27-20

ISAKMP Profile Bound to a Local Interface Configuration Example

The following example configures an ISAKMP profile bound to a local interface:

```
Router(config)# crypto isakmp profile profile1
Router(conf-isa-profile)# keyring keyring1
Router(conf-isa-profile)# match identity address 10.0.0.0 255.0.0.0
Router(conf-isa-profile)# local-address serial2/0
```

ISAKMP Keyring Bound to a Local Interface Configuration Example

The following example configures an ISAKMP keyring bound only to interface serial2/0:

```
Router(config)# crypto keyring keyring1
Router(conf-keyring)# local-address serial2/0
Router(conf-keyring)# pre-shared-key address 10.0.0.1
```

ISAKMP Keyring Bound to a Local IP Address Configuration Example

The following example configures an ISAKMP keyring bound only to IP address 10.0.0.2:

```
Router(config)# crypto keyring keyring1
Router(conf-keyring)# local-address 10.0.0.2
Router(conf-keyring)# pre-shared-key address 10.0.0.2 key
```

Certificate to ISAKMP Profile Mapping Configuration Examples

The following examples shows how to configure Certificate to ISAKMP Profile Mapping:

- Certificates Mapped to the ISAKMP Profile on the Basis of Arbitrary Fields Configuration Example, page 27-21
- Group Name Assigned to a Peer That Is Associated with an ISAKMP Profile Configuration Example, page 27-21

Certificates Mapped to the ISAKMP Profile on the Basis of Arbitrary Fields Configuration Example

The following example shows that whenever a certificate contains "ou = green," the ISAKMP profile "cert_pro" will be assigned to the peer:

```
crypto pki certificate map cert_map 10
 subject-name co ou = green
!
!
crypto isakmp identity dn
crypto isakmp profile cert_pro
   ca trust-point 2315
   ca trust-point LaBcA
   initiate mode aggressive
   match certificate cert_map
```

Group Name Assigned to a Peer That Is Associated with an ISAKMP Profile Configuration Example

The following example shows that the group "some_group" is to be associated with a peer that has been assigned an ISAKMP profile:

```
crypto isakmp profile id_profile
   ca trust-point 2315
   match identity host domain cisco.com

client configuration group some_group
```

Encrypted Preshared Key Configuration Example

The following example shows a configuration for which a type 6 preshared key has been encrypted. It includes the prompts and messages that a user might see.

```
Router(config)# password encryption aes
Router(config)# key config-key password-encrypt
New key:
Confirm key:
Router(config)#
0:46:40: TYPE6_PASS: New Master key configured, encrypting the keys with
the new master key
Router(config)# exit
```

IKE Aggressive Mode Initiation Configuration Examples

This section provides the following IKE Aggressive Mode Initiation configuration examples:

- IKE Aggressive Mode Initiation Hub Configuration Example, page 27-22
- IKE Aggressive Mode Initiation Spoke Configuration Example, page 27-22
- IKE Aggressive Mode Initiation RADIUS User Profile Example, page 27-23

IKE Aggressive Mode Initiation Hub Configuration Example

The following example shows how to configure a hub for a hub-and-spoke topology that supports IKE aggressive mode initiation using RADIUS tunnel attributes:

```
!The AAA configurations are as follows:
aaa new-model
aaa authorization network ike group radius
aaa authentication login default group radius
!
! The Radius configurations are as follows:
radius-server host 1.1.1.1 auth-port 1645 acct-port 1646
radius-server key rad123
!
! The IKE configurations are as follows:
crypto isakmp policy 1
authentication pre-share
!
! The IPSec configurations are as follows:
crypto ipsec transform-set trans1 esp-3des esp-sha-hmac
!
crypto dynamic-map Dmap 10
set transform-set trans1
!
crypto map Testtag isakmp authorization list ike
crypto map Testtag 10 ipsec-isakmp dynamic Dmap
!
interface Ethernet0
ip address 4.4.4.1 255.255.255.0
crypto map Testtag
!
interface Ethernet1
ip address 2.2.2.1 255.255.255.0
```

IKE Aggressive Mode Initiation Spoke Configuration Example

The following example shows how to configure a spoke for a hub-and-spoke topology that supports IKE aggressive mode initiation using RADIUS tunnel attributes:

```
!The IKE configurations are as follows:
crypto isakmp policy 1
authentication pre-share
!
! The IPSec configurations are as follows:
crypto ipsec transform-set trans1 esp-3des esp-sha-hmac
access-list 101 permit ip 3.3.3.0 0.0.0.255 2.2.2.0 0.0.0.255
!
! Initiate aggressive mode using Radius tunnel attributes
crypto isakmp peer address 4.4.4.1
set aggressive-mode client-endpoint user-fqdn user@cisco.com
set aggressive-mode password cisco123
```

```
!
crypto map Testtag 10 ipsec-isakmp
set peer 4.4.4.1
set transform-set trans1
match address 101
!
interface Ethernet0
ip address 5.5.5.1 255.255.255.0
crypto map Testtag
!
interface Ethernet1
ip address 3.3.3.1 255.255.255.0
```

IKE Aggressive Mode Initiation RADIUS User Profile Example

The following is an example of a user profile on a RADIUS server that supports the Tunnel-Client-Endpoint and Tunnel-Password attributes:

```
user@cisco.com Password = "cisco", Service-Type = Outbound
Tunnel-Medium-Type = :1:IP,
Tunnel-Type = :1:ESP,
Cisco:Avpair = "ipsec:tunnel-password=cisco123",
Cisco:Avpair = "ipsec:key-exchange=ike"
```

Call Admission Control for IKE Configuration Examples

The following examples shows how to configure Call Admission Control (CAC) for IKE:

- IKE Security Association Limit Configuration Example, page 27-23
- System Resource Limit Configuration Example, page 27-23

IKE Security Association Limit Configuration Example

The following example shows how to specify that there can be a maximum of 25 SAs before IKE starts rejecting new SA requests:

```
Router(config)# crypto call admission limit ike sa 25
```

System Resource Limit Configuration Example

The following example shows how to specify that IKE should drop SA requests when 90 percent of system resources are being used:

```
Router(config)# call admission limit 90
```

Dead Peer Detection Configuration Examples

The following examples show how to configure Dead Peer Detection (DPD):

- On-Demand DPD Configuration Example, page 27-24
- Periodic DPD Configuration Example, page 27-24
- DPD and Cisco IOS Keepalives with Multiple Peers in a Crypto Map Configuration Example, page 27-24

On-Demand DPD Configuration Example

The following example shows how to configure on-demand DPD messages. In this example, DPD messages will be sent every 60 seconds and every 5 seconds between retries if the peer does not respond:

```
Router(config)# crypto isakmp keepalive 60 5
```

Periodic DPD Configuration Example

The following example shows how to configure periodic DPD messages. In this example, DPD messages are to be sent at intervals of 10 seconds:

```
Router(config)# crypto isakmp keepalive 10 periodic
```

DPD and Cisco IOS Keepalives with Multiple Peers in a Crypto Map Configuration Example

The following example shows that DPD and IOS keepalives are used in conjunction with multiple peers in a crypto map configuration when IKE will be used to establish the security associations (SAs). In this example, an SA could be set up to the IPSec peer at 10.0.0.1, 10.0.0.2, or 10.0.0.3.

```
Router(config)# crypto map green 1 ipsec-isakmp
Router(config-crypto-map)# set peer 10.0.0.1
Router(config-crypto-map)# set peer 10.0.0.2
Router(config-crypto-map)# set peer 10.0.0.3
Router(config-crypto-map)# set transform-set txfm
Router(config-crypto-map)# match address 101
```

IPSec NAT Transparency Configuration Example

The following example shows how to enable NAT keepalives to be sent every 20 seconds:

```
crypto isakmp policy 1
authentication pre-share
crypto isakmp key 1234 address 56.0.0.1
crypto isakmp nat keepalive 20
!
!
crypto ipsec transform-set t2 esp-des esp-sha-hmac
!
crypto map test2 10 ipsec-isakmp
set peer 56.0.0.1
set transform-set t2
match address 101
```