IPsec VPN Configuration and Operation Using the SmartEdge OS CLI

SYSTEM ADMINISTRATOR GUIDE

2/1543-CRA 119 1170/1-V1 Uen G

Copyright

© Ericsson AB 2009–2011. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

SmartEdge	is a registered trademark of Telefonaktiebolaget LM Ericsson.
NetOp	is a trademark of Telefonaktiebolaget LM Ericsson.

Contents

1 Introduction

An Internet Protocol Security (IPsec) Virtual Private Network (VPN) consists of protected tunnels between pairs of gateways. A gateway, also referred to as a peer, is a node that functions as an endpoint for a protected tunnel that traverses an unsecured routing cloud. A gateway protects the traffic communicated by networks at each end of the tunnel that are outside the routing cloud; these networks are known as protected networks. You can use a SmartEdge® router provisioned with an Advanced Services Engine (ASE) card that is configured to provide the Security Service as an IPsec VPN gateway. For information on provisioning an ASE card and configuring the Security Service, see Reference [6].

1.1 Scope

This document describes how to use the SmartEdge OS to add and configure all of the elements required for an IPsec VPN tunnel endpoint to the configuration of a SmartEdge router, including Internet Key Exchange (IKE) and IPsec configuration. This document also provides information on how to monitor IPsec, example configurations, and log messages.

1.2 Target Groups

This document is intended for network planners responsible for the design of advanced network services that use the SmartEdge router and for operators of the SmartEdge OS responsible for entering the configuration on individual SmartEdge routers.

2 IPsec VPN Introduction

An IPsec VPN is a set of IPsec tunnels that securely isolates the traffic between the participating peer devices from any other traffic traversing the same devices.

An IPsec tunnel consists of two endpoints that successfully negotiate a secure connection using the IKEv1 or IKEv2 protocol. The IKEv2 protocol is more robust, and offers improved performance and greater security than the IKEv1 protocol. The two protocols do not interoperate, but both have enough of the header format in common for each to unambiguously run over the same UDP port.

Each ASE card can support up to 8,000 IPsec tunnel sessions using IKEv1 and/or IKEv2 signaling, up to a total of 32,000 sessions for the SmartEdge router.

IPsec tunnel endpoints are configured independently on each participating peer in an IPsec VPN. The simplest configuration, site-to-site IPsec VPN, is between two peers. On each peer, you configure a local IPsec tunnel endpoint that points to the remote endpoint on the other peer.

A hub-and-spoke configuration consists of many IPsec tunnel endpoints on one hub peer, each of which has the same local identity and points to a different spoke peer, and only one IPsec tunnel endpoint on each spoke peer that points back to the hub peer. By adding more tunnel endpoints to each of the peers, you can configure anything from a partial to a fully-meshed IPsec VPN.

IPsec tunnels can be configured in two modes:

- Circuit-based (the default)

  The circuit-based mode directs incoming clear tunnel traffic to the line card before and after forwarding it to the ASE card for encryption and then to the egress port. Circuit-based mode allows access to all the circuit-based resources available on line cards. It is required to use the supported dynamic routing protocols, OSPF and RIP, which automatically install IP routes to the IPsec tunnel as soon as the tunnel is operational.

- Economical

  The economical mode directs incoming clear tunnel traffic to the line card before forwarding it to the ASE card for its encryption services and then directly to the egress port. Economical mode makes more efficient use of the line card, accessing only the routing capabilities of the card. Economical mode is preferable for IPsec tunnels that do not use dynamic routing protocols. Economical mode must be explicitly specified when you configure an IPsec tunnel.

A secure connection for an IPsec tunnel is established after negotiating a Service Association (SA) at each endpoint. SAs can be established in two ways:

- Automatically, by negotiating the SAs using the IKEv1 or IKEv2 protocol. There are two variations:

  - Static Auto Key—SAs are negotiated between two known local and remote end points.

  - On-Demand Auto Key—Same as Auto Key, except that the remote peer IP address is unknown at configuration, and IPsec tunnels are dynamically created on demand, based on signaling from the remote peer. Multiple remote peers can connect to the same on-demand auto key configuration using the same local IP address.

  Network administrators on both sides of the connection configure complementary IKE proposals and IKE policies to specify the attributes for the IKE negotiations.

- Manual Key—Network administrators on both sides of the connection define the SAs manually, and specify the SAs to be used when configuring a manual key IPsec tunnel.

Auto key configuration is more secure and removes the potential for error compared to the manual technique. You can deploy IKE using preshared keys to authenticate both parties. With manual keys, parties at both ends of a tunnel configure the security parameters. This technique is straightforward for small, static networks where management and distribution of keys is relatively simple. However, manual key-based configurations create potential for security breaches.

Auto key configuration also supports traffic-selector guided route addition. Each endpoint learns the addresses of the networks being secured behind the remote endpoint from traffic selectors that are negotiated as part of the IKE negotiations used to establish the IPsec SAs. Traffic selectors specify the IP address, protocol, or ports secured by each IPsec SA, and exist in pairs: source traffic selector and destination traffic selector. An endpoint can negotiate several IPsec SAs, each securing a different pair of traffic selectors. The same traffic selectors are seen reversed from the perspective of each peer. After an endpoint learns the addresses of the networks secured behind each peer, traffic-selector guided route addition can be enabled to automatically add routes from the local endpoint to each protected network. These point to the IPsec tunnel that connects the two endpoints.

3 IPsec VPN Prerequisites

Before you can create an IPsec VPN, the two peers of an IPsec tunnel must be logically connected; all the participating peers must be set up and connectivity between all the gateways must be established. For each SmartEdge router that functions as a peer endpoint for an IPsec tunnel, you configure the local port or circuit termination by using the SmartEdge OS CLI or the NetOp™ client. Configuring the connectivity for unmanaged remote peers is a separate activity beyond the scope of this document. Consult the vendor documentation for these devices.

To configure the connectivity on a SmartEdge router that functions as a peer endpoint for an IPsec tunnel: do the following:

- Identify the routing context in which you are configuring the IPsec tunnel and define it, if it does not exist.
- Enable the routing context for Security Service; see *Enabling a Context to Provide an ASE-Based Service* in Reference [6].
- In the routing context in which you are configuring the IPsec tunnel, configure the following IP interfaces and bind them appropriately:
  - Peer interface—An interface to bind to the link between the two peers.
  - Traffic interface—An interface to bind to the physical link between the SmartEdge router and the protected network.
  - Tunnel interface—An interface to statically bind to the logical IPsec tunnel. Create one interface for each IPsec tunnel.

  For information on how to configure an interface or bind a port or circuit to an interface by using the SmartEdge OS, see Reference [1], and by using the NetOp Element Management System (EMS), see Reference [2].
- Configure the gateway interface, a loopback interface whose IP address will be used as the local endpoint of the tunnel. This interface does not need to be configured in the same routing context as the peer, traffic, and tunnel interfaces.
- Configure IP routes to point to the IPsec tunnel interface. There are three ways you can configure IP routes to point to the IPsec tunnel interface:
  - Adding static routes explicitly. For information on how to configure a static IP route by using the SmartEdge OS; see Reference [3].
  - Running dynamic routing protocols such as Open Shortest Path First (OSPF) and RIP, which install routes to a tunnel as soon as the tunnel is established; see Reference [4] and Reference [5]. Available only for IPsec tunnels configured in circuit-based mode.

– Enabling traffic-selector-based route addition when you configure the IPsec tunnel, which uses the range of addresses of the secured networks of the remote peer learned during SA negotiation to dynamically add IP routes pointing to the tunnel when the tunnel comes up and delete them when the tunnel goes down; see Section 9 on page 43.

4 Site-to-Site Connections Examples

You can create IPsec tunnels between two SmartEdge routers or between one managed SmartEdge router and an unmanaged device (referred to as an extranet device), such as Customer Premises Equipment (CPE) or remote special-purpose server.

Figure 1 shows the physical links for an IPsec VPN that must exist between two managed SmartEdge routers and between the two SmartEdge routers and the protected networks at each end of the IPsec tunnel.

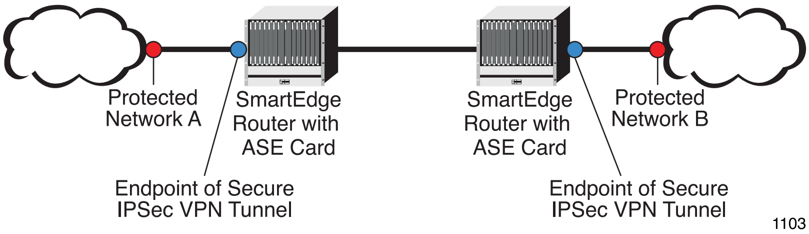

Figure 1 An IPsec VPN Between Two Managed SmartEdge Routers

Figure 2 shows the physical links for an IPsec tunnel that must exist between the NetOp EMS host, a managed SmartEdge router, and an extranet device as well as the SmartEdge router and the protected network at its end of the IPsec tunnel.

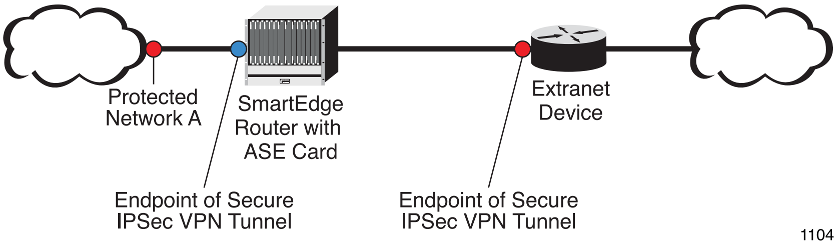

Figure 2 An IPsec Tunnel Between a Managed SmartEdge Router and an Extranet Device

You can use these two basic configurations to implement a variety of individual secure site-to-site IPsec tunnels.

4.1 Connecting Remote Protected Networks

An IPsec tunnel between two managed SmartEdge routers protects traffic across the untrusted network between the routers. This type of IPsec tunnel allows a service provider to offer protected traffic support between the endpoints that it manages.

For example, Figure 3 shows an IPsec VPN that connects two protected networks at each end of the tunnel for a multisite business customer (a total of four protected networks). The customer trusts the traffic between its premises and the SmartEdge routers, but does not trust the traffic between the SmartEdge routers managed by the service provider (also known as Provider Equipment [PE]).

To secure the traffic of the business customer between its SmartEdge routers, the service provider uses an IPsec tunnel. This ensures that all the traffic for the business customer between the two endpoints the service provider manages is protected.

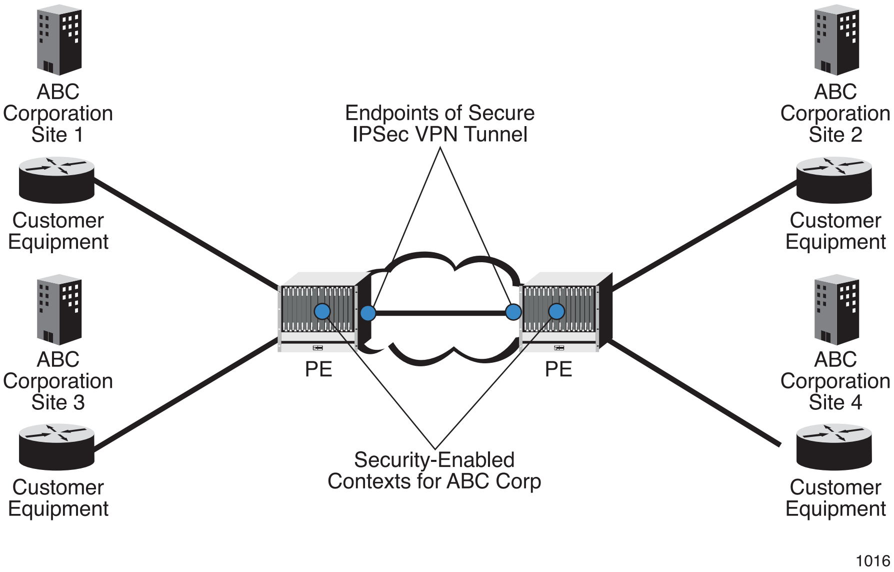

Figure 3 An IPsec Tunnel That Protects Traffic Between Two SmartEdge Routers

In this scenario, the connections between the equipment on the premises of the customer and the SmartEdge routers of the service provider, as well as the connection between the SmartEdge routers, must already be configured. To protect the customer traffic between the two managed SmartEdge routers, the service provider must configure both endpoints of the secured IPsec tunnel.

4.2 Securing the Last Mile

An IPsec tunnel between a managed SmartEdge router and a CPE device protects traffic on the last mile. This type of IPsec tunnel allows a service provider to offer protected services between an endpoint it manages and one that it does not.

For example, Figure 4 shows four IPsec tunnels that connect the four protected networks to two SmartEdge routers for a multisite business customer. The customer expects the service provider to protect the traffic between its premises and the SmartEdge routers.

To secure this traffic, the service provider uses four IPsec tunnels to ensure all the traffic between each CPE device at the customer site and the SmartEdge router it manages is protected.

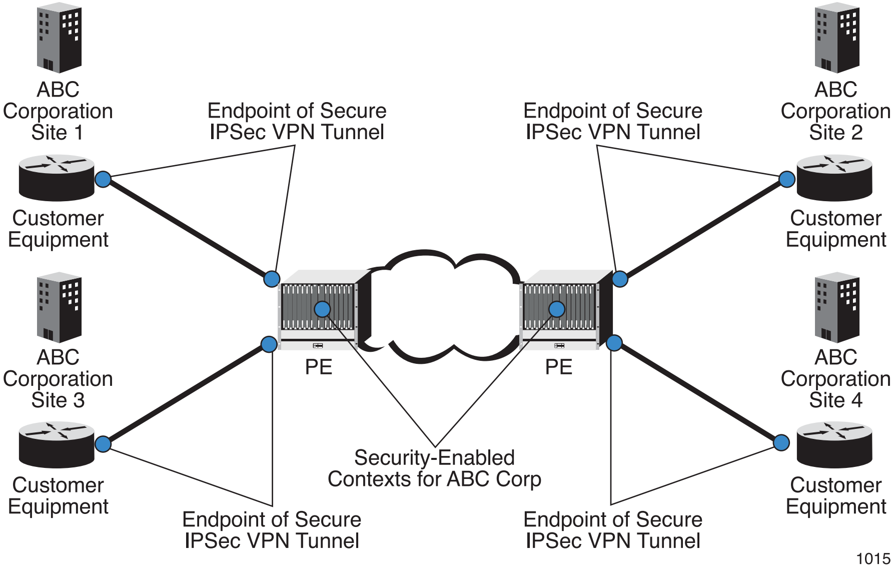
ABC Corporation Site 1
Endpoint of Secure IPSec VPN Tunnel
Endpoint of Secure IPSec VPN Tunnel
ABC Corporation Site 2
Customer Equipment
Customer Equipment
ABC Corporation Site 3
PE
PE
ABC Corporation Site 4
Security-Enabled Contexts for ABC Corp
Customer Equipment
Customer Equipment
Endpoint of Secure IPSec VPN Tunnel
Endpoint of Secure IPSec VPN Tunnel
1015

Figure 4 An IPsec Tunnel That Protects Traffic Between Customer Equipment and a SmartEdge Router

In this scenario, the connections between the equipment on the customer premises and the service provider SmartEdge routers, as well as the connection between the SmartEdge routers, must already be configured. To protect the customer traffic between its premises and the service provider SmartEdge router, on each IPsec tunnel the service provider must configure the endpoint on the SmartEdge router and identify the peer endpoint on the extranet device on the customer premises.

4.3 Encrypting Local Stack Packets

Service providers need to encrypt their own data to send to customers because the data has encryption requirements associated with it. Common requirements include organizational best practices, risks associated with the data, regulatory demands, and privacy concerns.

For example, Figure 5 shows three IPsec tunnels that protect the traffic of a service provider on connections between its own equipment in the network.

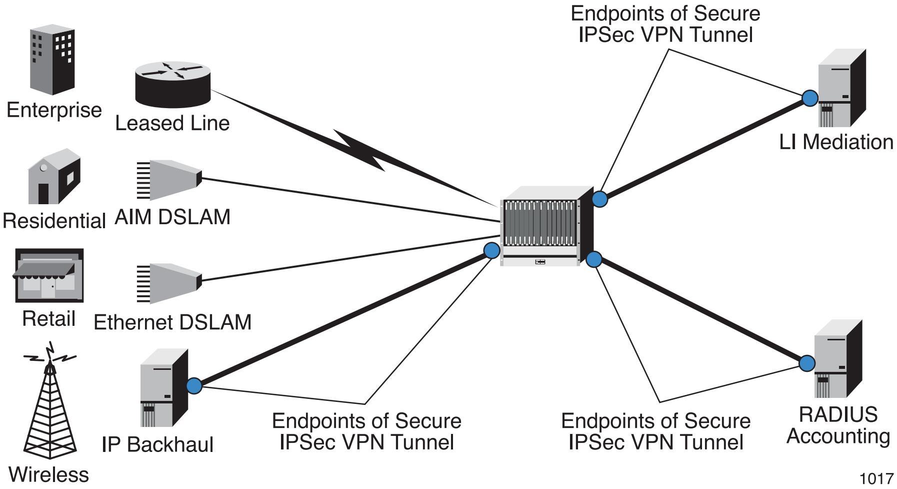

Figure 5 Service Provider Data Requiring Encryption

In this scenario, the service provider is protecting its own data, and the connections between all equipment involved in communicating the encrypted data must already be configured. To protect its own traffic, the service provider must also configure individual IPsec tunnels and identify the appropriate endpoints.

5 IPsec VPN Setup and Activation Using IKE

Both IKEv1 and IKEv2 protocols are supported.

The minimal matching requirements for both endpoints of an IPsec tunnel to successfully complete all negotiations and the setup to become active are as follows:

- IKE proposal—Each peer must have at least one referenced matching IKE proposal. In the IKE proposal on each peer, the values listed below must match.

Value	**IKEv1**	**IKEv2**
Encryption Algorithm	X	X
DH Group	X	X
Authentication Algorithm	X	n/a
Authentication method	X	
Lifetime value	X	n/a
Pseudo Random Function	n/a	X

- IKE policy—Each peer must have at least one referenced matching IKE policy. In the IKE policy on each peer, the values listed below must match:

Value	**IKEv1**	**IKEv2**
IKE Proposal	X	X
Preshared Key(1)	X	
Digital signatures(2)	X	
Key exchange mode	X	X

(1) If authentication method set in the referenced IKE proposal is `pre-shared-key`*.*

(2) If authentication method in the referenced IKE proposal is `rsa-signature`*.*

In addition, the connection type on each peer must be compatible (at least one initiator and one responder).

If certificate authentication is specified for an IKEv1 proposal or an IKEv2 policy, you can also use the `validate-certificate-identity` command to check that the ID received in the self-certificate from the remote peer of any incoming tunnel matches the remote peer ID configured for that tunnel on the local peer.

- IPsec proposal—Each peer must have at least one configured matching IPsec proposal.
- IPsec policy:

  - At least one referenced IPsec proposal on each peer must match.
  - The Perfect Forward Secrecy (PFS) DH group settings must match.

- IPsec SA (only configured for use by manual key IPsec tunnels):
  - Bidirectional SAs are configured with all attributes matching.
  - One-directional SAs must have matching security protocol attributes.
- IPsec Access Control List (ACL)—The rules in the IPsec ACL referenced by each peer must match.
- Within each IPsec tunnel endpoint configuration:
  - The local and remote addresses or Fully Qualified Domain Names (FQDNs) and the local and remote IKE identities must match.
  - The referenced IKE policy must be consistent with the IKE policy referenced by the peer endpoint.
  - The endpoint of the logical IPsec tunnel must be statically bound to an interface.
  - IPsec QoS Policies are local to the node and do not need to match at each endpoint. However, it is good practice to have a consistent set of IPsec QoS policies defined on all nodes in the network.
  - The local endpoint of an on-demand auto key IPsec tunnel uses a tunnel-specific IPsec profile to specify how its traffic should be handled. Although the IPsec profiles at each endpoint need not match, an IPsec profile references up to eight IPsec policies and, optionally up to the same number of IPsec ACLs. Any IPsec policy or IPsec ACL referenced by one peer of an on-demand auto key tunnel must match on the other peer according to the criteria described above.

6 IKE Configuration

The IPsec VPN application supports the dynamic negotiation and exchange of secure keys and SAs using an IKE protocol. This is known as the auto key technique and it allows the use of certificates or preshared keys to authenticate both parties. Compared to the manual technique, the auto key technique is more secure and removes the potential for error. You can use either the IKEv1 or IKEv2 protocol. IKE configuration consists of:

- Configuring the Public Key Infrastructure (PKI) on the node if digitally-signed certificates are used to authenticate both parties.
- Configuring global IKE proposals, using either IKEv1 or IKEv2 protocol, which define parameters used to negotiate and exchange the keys and SAs used to set up IKE SAs.
- Optionally enabling Dead Peer Detection (DPD) in each security-enabled context, which provides an efficient way to detect when remote peers become unavailable.
- Configuring IKE policies in each security-enabled context, using either IKEv1 or IKEv2 protocol, which provide specific attributes to use when establishing IKE SAs.

Compared to the earlier IKEv1 protocol, the IKEv2 protocol provides the following features:

- Allows the SA at each endpoint to enforce its own lifetime policy, which enables the SAs to establish new SAs (child SAs) when the lifetime of the current SAs is about to expire. This is known as rekeying.
- Negotiates a Pseudo-Random Function (PRF) used to construct the keying material used for all of the cryptographic algorithms used in any SA.
- Enables each endpoint to independently authenticate the remote peer, which allows one endpoint to authenticate remote peers using preshared keys and the other using RSA certificates.
- Allows traffic selectors to be negotiated on the receiver side; for example, a remote peer can propose forwarding all traffic, and the local peer can be configured to negotiate receipt of a subset of the proposed traffic.

6.1 Configuring PKI

You can use digitally-signed certificates to authenticate the identity of the two IPsec tunnel endpoints establishing a secure connection during the set up of an IPsec tunnel by configuring `authentication rsa-signature` in an IKEv1 proposal or an IKEv2 policy.

Digital signatures can only be used if you have configured support for the Public Key Infrastructure) PKI on each peer. PKI utilizes public key cryptography, which uses a pair of keys at both ends: a public key, known at both ends, and a private key, known only at one end. The two keys, called a key pair, are mathematically related. A private key digitally signs the authentication payload and the accompanying public key verifies the signatures.

Public key cryptography is stronger than shared (or symmetric) key cryptography, but consumes substantial processing resources. Therefore, it is only used for the exchange of authentication information during IPsec tunnel setup. Once the identities of both peers is authenticated, the preshared keys configured in the IKE policy are used to secure data.

Before you can configure PKI on a SmartEdge router to authenticate the identities of IPsec tunnel endpoints you must obtain a trusted certificate from a Certificate Authority (CA). The trusted certificate contains the public key of the CA and authenticates the identity of the SmartEdge router to the CA. Obtaining a trusted certificate from a CA is outside the scope of this document. The trusted certificate enables you to request the self certificates from the CA that are used to authenticate the identities of the IPsec tunnel endpoints. The procedures in the following sections assume that a trusted certificate from a CA is available.

PKI configuration is a manual process that must be completed in the context which the local endpoints of IPsec tunnels whose endpoint identities are authenticated with certificates are configured:

- Import the trusted certificate.

  The trusted certificate provides the CA public key that is the SmartEdge router uses to authenticate itself when requesting self certificates from the CA.

- Create a key pair for a self certificate on the SmartEdge router. Alternatively, you can import a key pair created by a CA from a file in PEM format.

- Request signed self-certificates.

  The result of the certificate request is a file in PEM format containing the unsigned self-certificate which must be manually submitted to a CA by a system administrator. The CA uses the private key it owns that is associated with the CA public key provided by the SmartEdge router and:

  - Calculates the hash of the certificate request and encrypts the result.

  - Encrypts the hash with its private key.

- Signs the self-certificate by adding the encrypted hash.
- Returns a PEM format file containing the signed self-certificate.

Submission of a certificate request to a CA and receipt of a signed self-certificate from a CA is outside the scope of this document.

- Import signed self-certificates.

6.1.1 Configuration Tasks

Configure PKI in each context in which you configure the local endpoints of IPsec tunnels whose endpoint identities are authenticated with certificates. Navigate to the routing context and configure PKI in global configuration mode.

To configure PKI in a security-enabled context:

1. Import the trusted certificate, which must be a file in PEM format, that you obtained from a CA:

   ```
   #import pki certificate trusted rsa file file-name
   ```

2. Create the public and private key pair:

   ```
   #add pki key-pair key-pair-name rsa key size
   ```

   Alternatively, you can import a private and public key pair generated by a CA from a file in PEM format into the context:

   ```
   #import pki key-pair key-pair-name type file file-name
   ```

3. Request a self-certificate:

   ```
   #add pki certificate-request rsa request-name file
   file-name
   ```

   You are then prompted to enter the following information (responses are not shown in this example):

   ```
   Enter the private key identifier:
   Enter the Common name:
   Enter the Organization Unit Name:
   Enter the Organization Name:
   Enter the City Name:
   Enter the State Name:
   Enter the Postal Code:
   Enter the Country Code:
   % SUBJECT ALTERNATIVE NAMES
   Enter the IPv4 Address:
   Enter the IPv4 Address:
   Enter the IPv6 Address:
   Enter the IPv6 Address:
   Enter the Domain Name:
   ```

```
Enter the Domain Name:
Enter the Email Id:
Enter the Email Id:
Do you want to configure Key Usage Attributes(y/n):
DigitalSignature (y/n):
NonRepudiation (y/n):
KeyEncipherment(y/n):
DataEncipherment (y/n):
KeyAgreement (y/n):
EncipherOnly (y/n):
DecipherOnly (y/n):y
```

Note: Some items of information are entered twice for confirmation purposes.

4. Complete the following tasks:

   - Forward the certificate request to the CA. The CA returns a file containing the signed self-certificate.

   - After you receive the file containing the signed self-certificate from the CA, save it in a locally accessible location, such as the flash partition on the SmartEdge router.

   The procedures to complete the tasks in this step are outside the scope of this document.

5. Import the self-certificate:

   ```
   #import pki certificate self rsa key-pair key-pair-name
   file file-name
   ```

6. A certificate is identified by the value of its handle. Use the following command to list all certificates and their handles:

   ```
   #show pki certificate self rsa
   ```

   To delete a certificate, you must provide its handle:

   ```
   #remove pki certificate handle
   ```

6.1.2 Configuration Examples

The following example imports the trusted certificate in the file `cacert.pem` from the flash partition on the SmartEdge router into context local:

```
[local]Redback#import pki certificate trusted rsa file
/flash/cacert.pem
```

The following example creates a key pair `keytest1` with the key type `rsa` and a key length of `512` bits in context local:

```
[local]Redback#add pki key-pair keytest1 rsa 512
```

The following example creates the request file `test1` in the flash partition on the SmartEdge router for a self-certificate that you can submit to a CA using the key pair test1 generated in the previous example:

```
[local]Redback#add pki certificate-request rsa test1 file
test1
```

Prompts for information accepted in the file appear one at a time. In the following example, no information is provided for subject alternative names.

```
Enter the private key identifier:keytest1

Enter the Common name:An Operator

Enter the Organization Unit Name:Provider

Enter the Organization Name:Service

Enter the City Name:My City

Enter the State Name:XY

Enter the Postal Code:98765

Enter the Country Code:ZZ

% SUBJECT ALTERNATIVE NAMES

Enter the IPv4 Address:

Enter the IPv4 Address:

Enter the IPv6 Address:

Enter the IPv6 Address:

Enter the Domain Name:

Enter the Domain Name:

Enter the Email Id:

Enter the Email Id:

Do you want to configure Key Usage Attributes(y/n):n
```

The following example imports the digitally-signed self certificate for the key pair `keytest1` from the file `newcert1.pem` in the flash partition on the SmartEdge router into context local:

```
[local]Redback#import pki certificate self rsa key-pair
keytest1 file /flash/newcert1.pem
```

6.2 Configuring IKE Proposals

An IKE proposal defines the parameters used to negotiate an IKE SA: the algorithm used to encrypt the signaling traffic, the algorithm used to authenticate both parties to establish the VPN connection, and the Diffie-Hellman (DH) group to use.

An IKE proposal specifies:

- DH group: 1, 2, 5, or 14
- Authentication algorithm: hmac-md5-96, hmac-sha1-96
- Authentication method (IKEv1 proposals only): preshared-key, rsa-signature
- Encryption algorithm: aes-128-cbc, aes-192-cbc, aes-256-cbc, des-cbc, 3des-cbc
- Pseudo random function (IKEv2 proposals only): hmac-md5, hmac-sha1, aes-128-xcbc
- SA lifetime (IKEv1 proposals only): 300 to 99,999,999 seconds

You can configure proposals for either the IKEv1 or IKE2 protocol:

- Use the `ike proposal` command to configure an IKEv1 proposal.
- Use the `ike2 proposal` command to configure an IKEv2 proposal.

A few settings for an IKE proposal differ between the two supported IKE protocols. An IKEv1 proposal negotiates the peer authentication method (either preshared key or RSA signature) and the SA lifetime for both endpoints (and therefore does not support rekeying of SAs) . An IKEv2 proposal negotiates an algorithm for a pseudo random function (therefore supports rekeying of SAs) and does not negotiate SA lifetime settings (authentication method or SA lifetime). Both IKEv1 and IKEv2 proposals negotiate the DH group, and the authentication and encryption algorithms.

A newly created IKE proposal is automatically configured with the default values listed in Table 1.

Table 1 Default Values for an IKE Proposal

Name	Value	IKE Proposal Type
DH group	1	Both
Authentication algorithm	hmac-sha1-96	Both
Authentication method	pre-shared-key	IKEv1 Proposal
Encryption algorithm	aes-128-cbc	Both

Name	Value	IKE Proposal Type
Pseudo Random Function	hmac-sha1	IKEv2 Proposal
SA lifetime	86,400 seconds (one day)	IKEv1 Proposal

6.2.1 Configuration Tasks

To configure an IKE proposal:

1. Create an IKE proposal in global configuration mode:

   - To create an IKEv1 proposal:

     ```
     (config)#ike proposal name
     ```

   - To create an IKEv2 proposal:

     ```
     (config)#ike2 proposal name
     ```

2. In IKE proposal configuration mode:

   - Configure the description of the IKE proposal:

     ```
     (config-ike-proposal)#description string
     ```

   - Configure the DH group for IKE key exchanges:

     ```
     (config-ike-proposal)#dh-group dh-group
     ```

   - Configure the authentication algorithm:

     ```
     (config-ike-proposal)#authentication algorithm
     algorithm
     ```

   - For an IKEv1 proposal only, configure the peer authentication method:

     ```
     (config-ike-proposal)#authentication {pre-shared-key
     |rsa-signature}
     ```

     If `pre-shared-key` is configured, the pre-shared key specified in the IKEv1 policy that references this proposal is used for authenticating the identity of each peer during the set up of the IPsec tunnel. If `rsa-signature` is configured, digitally signed authentication data from a CA is used for authenticating the identity of each peer during the set up of the IPsec tunnel. To digitally sign authentication data you must configure PK and install a trusted certificate from a CA and self certificates signed by the CA; see Section 6.1 on page 16.

   - Configure the encryption algorithm:

```
(config-ike-proposal)#encryption algorithm algorithm
```

- For an IKEv1 proposal only, configure the lifetime for IKE SAs:

  ```
  (config-ike-proposal)#lifetime seconds seconds
  ```

- For an IKEv2 proposal only, configure the pseudo random function for constructing the keying material for all cryptographic algorithms used in SAs:

  ```
  (config-ike2-proposal)#pseudo-random-function
  algorithm
  ```

3. Commit the transaction.

6.2.2 Configuration Examples

The following example shows how to configure the `IKE_Prop1` IKEv1 proposal:

```
[local]Redback(context)#ike proposal IKE_Prop1
[local]Redback(config-ike-proposal)#description IKE-Proposal-1
[local]Redback(config-ike-proposal)#dh-group 2
[local]Redback(config-ike-proposal)#authentication algorithm
  hmac-md5-96
[local]Redback(config-ike-proposal)#encryption algorithm aes-192-cbc
[local]Redback(config-ike-proposal)#lifetime seconds 43200
```

Example 1 Configuration for an IKEv1 Proposal

The following example shows how to configure the `IKE2_Prop1` IKEv22 proposal:

```
[local]Redback(context)#ike2 proposal IKE2_Prop1
[local]Redback(config-ike-proposal)#description IKE2-Proposal-1
[local]Redback(config-ike-proposal)#dh-group 14
[local]Redback(config-ike-proposal)#authentication algorithm hmac-md5-96
[local]Redback(config-ike-proposal)#encryption algorithm aes-192-cbc
[local]Redback(config-ike-proposal)#pseudo-random-function hmac-md5
```

Example 2 Configuration of an IKEv2 Proposal

6.3 Enabling and Disabling IKE Keepalive

In each security-enabled context, you can enable the sending of Dead Peer Detection (DPD) messages to IKE peers. By default, the sending of DPD messages is disabled.

6.3.1 Configuration Tasks

To enable the sending of DPD messages to IKE peers, enter the following command in context configuration mode:

```
(config-ctx)#ike keepalive
```

The no form of the command disables the sending of DPD messages to IKE peers.

6.3.2 Configuration Examples

The following example shows how to enable the sending of DPD messages to IKE peers:

```
[local]Redback(config-ctx)#ike keepalive
```

The following example shows how to disable the sending of DPD messages to IKE peers:

```
[local]Redback(config-ctx)#no ike keepalive
```

6.4 Configuring IKE Policies

An IKE policy provides settings to apply to IPsec tunnels when negotiating an IKE SA with the remote peer. An IKE policy is created in the context that the local endpoint of the IPsec tunnel that refers to it. You can configure IKE policies for either the IKEv1 or IKEv2 protocol. An IKEv1 policy can only reference IKEv1 proposals and an IKE v2 policy can only reference IKEv2 proposals. The settings in an IKE policy are not negotiable.

An IKE policy specifies:

- Connection type:
  - Initiator only
  - Responder only
  - Both
- Local identity.
- Type of key exchange to use:
  - Aggressive mode
  - Main mode
- Authentication method (IKEv2 polices only):
  - Preshared key (PSK)
  - RSA signature
- Allocation Address to identify the AAA server that is the source of address allocation for remote access clients using the policy (IKEv2 policies only).
- The SA lifetime (IKEv2 policies only)
- Preshared key to use with the remote peers defined in the IKE policy if preshared keys are specified as the authentication method. You can provide an unencrypted key, an encrypted key, or specify that the preshared key information is fetched from AAA.
- IKE proposals to use. An IKE policy can reference multiple IKE proposals.

You can configure policies for either the IKEv1 or IKE2 protocol:

- Use the `ike policy` command to configure an IKEv1 policy.
- Use the `ike2 policy` command to configure an IKEv2 policy.

The configurable values for an IKE policy differ between the two supported IKE protocols. An IKE V1 policy supports two key exchange modes, does not set the local SA lifetime, and does not support address allocation for remote access clients. An IKEv2 policy does not support alternative key exchange modes, sets the local SA lifetime, and supports AAA address allocation.

A newly created IKE policy is automatically configured with the default value listed in Table 2.

Table 2 Default Values for an IKE Policy

Name	Value	IKE Policy Type
Connection type	both	Both
Authentication	pre-shared-key	Both
Key exchange mode	main	IKEv1 policy
SA lifetime	86,400 seconds (one day)	IKEv2 policy

Figure 6 illustrates the parameters that can be configured for an IKE policy, and that a policy can include multiple IKE proposals.

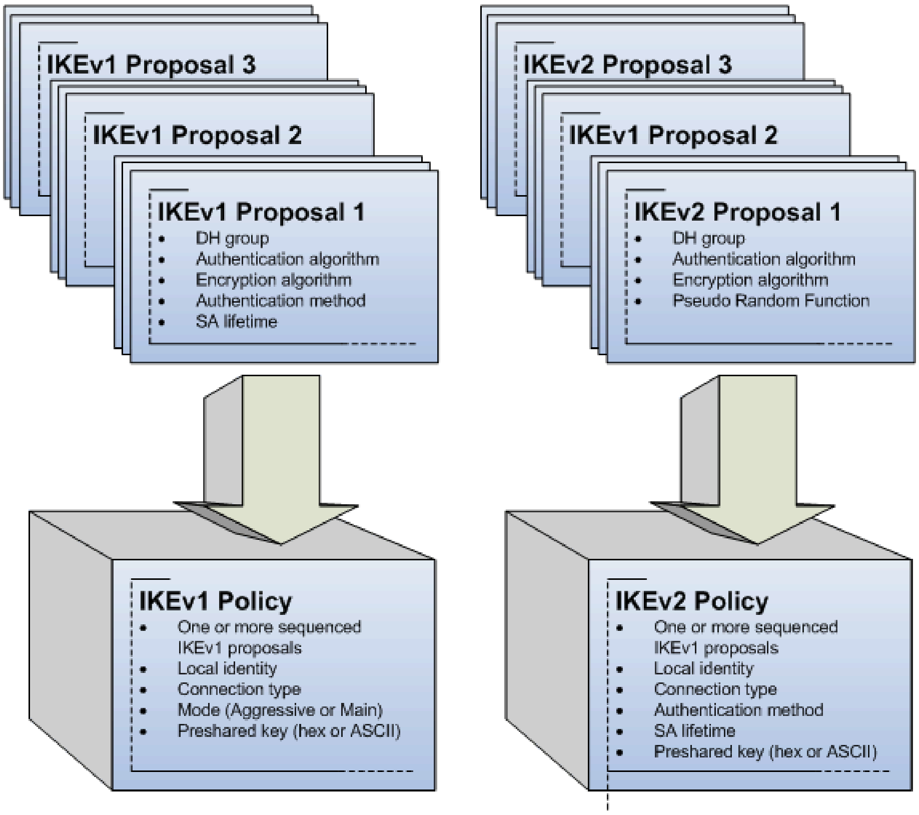

Figure 6 IKE Policy Parameters

6.4.1 Configuration Tasks

To configure an IKE policy:

1. Create an IKE policy in context configuration mode:

   - To create an IKEv1 policy:

     ```
     (config-ctx)#ike policy name
     ```

   - To create an IKEv2 policy:

     ```
     (config-ctx)#ike2 policy name
     ```

2. In IKE policy configuration mode:

   - Configure the description of the IKE policy:

     ```
     (config-ike-policy)#description string
     ```

   - Configure the connection type group for IKE key exchanges:

     ```
     (config-ike-policy)#connection-type {initiator-only|
     responder-only|both}
     ```

   - Required only if the connection type is `responder-only`. Specify that address allocation from an AAA server is used:

     ```
     (config-ike-policy)#address-allocation aaa
     ```

   - For an IKEv2 policy only, configure the peer authentication method:

     ```
     (config-ike-policy)#authentication {pre-shared-key|
     rsa-signature}
     ```

     If `pre-shared-key` is configured, the pre-shared key specified in the IKEv1 policy that references this proposal is used for authenticating the identity of each peer during the set up of the IPsec tunnel. If `rsa-signature` is configured, digitally signed authentication data from a CA is used for authenticating the identity of each peer during the set up of the IPsec tunnel. To digitally sign authentication data you must configure PK and install a trusted certificate from a CA and self certificates signed by the CA; see Section 6.1 on page 16.

   - Optionally, if the authentication specified in the IKEv1 proposal or in this IKEv2 proposal is `rsa-signature`, you can validate the identity of the remote peer in the self-certificate provided by the remote peer against the identity of the remote peer configured locally:

     ```
     (config-ike-policy)#validate-certificate-identity
     ```

   - Specify the identity to use as the local IKE identity for all IPsec tunnels that use this policy. Enter the IP address or FQDN of the loopback interface configured on the local SmartEdge router as the identity:

```
(config-ike-policy)#identity local fqdn fqdn-string
value
```

- Configure the mode to use for key exchange:

  ```
  (config-ike-policy)#mode mode
  ```

- Configure the local preshared key:

  ```
  (config-ike-policy)#pre-shared-key {hex hex-value|A
  SCII-value|use-aaa}
  ```

  The `use-aaa` keyword is only applicable to on-demand auto key IPsec tunnels; see Section 7 on page 29.

- For an IKEv2 policy only, configure the lifetime for IKE SAs:

  ```
  (config-ike-proposal)#lifetime seconds seconds
  ```

- Add up to 16 IKE proposals, each with a unique sequence number:

  ```
  (config-ike-policy)#seq sequence-number proposal name
  ```

  An IKEv1 policy can only reference IKEv1 proposals. An IKEv2 policy can only reference IKEv2 proposals.

3. Commit the transaction.

6.4.2 Configuration Examples

The following example shows how to configure the **`IKE_Pol1`** IKEv1 policy:

```
[local]Redback(config-ctx)#ike policy IKE_Pol1
[local]Redback(config-ipsec-proposal)#description IKE-Policy-1
[local]Redback(config-ike-policy)#connection-type initiator-only
[local]Redback(config-ike-policy)#identity local 30.0.1.3
[local]Redback(config-ike-policy)#preshared-key hex 0x4d794865785061353577307264
[local]Redback(config-ike-policy)#mode aggressive
[local]Redback(config-ike-policy)#seq 10 proposal IKE_Prop1
[local]Redback(config-ike-policy)#seq 20 proposal IKE_Prop2
```

Example 3   Configuration of an IKEv1 Policy

The following example shows how to configure the **`IKE2_Pol1`** IKEv2 policy:

```
[local]Redback(config-ctx)#ike policy IKE2_Pol1
[local]Redback(config-ipsec-proposal)#description IKE2-Policy-1
[local]Redback(config-ike-policy)#connection-type responder-only
[local]Redback(config-ike-policy)#identity local 30.0.1.3
[local]Redback(config-ike-policy)#preshared-key hex 0x4d794865785061353577307264
[local]Redback(config-ike-policy)#authentication rsa-signature
[local]Redback(config-ike-policy)#validate-certificate-identity
[local]Redback(config-ike-policy)#address-allocation aaa
[local]Redback(config-ike-policy)#lifetime 43200
[local]Redback(config-ike-policy)#seq 10 proposal IKE2_Prop1
[local]Redback(config-ike-policy)#seq 20 proposal IKE2_Prop2
```

Example 4   Configuration of an IKEv2 Policy

7 AAA Configuration

You can use Authentication, Authorization, and Accounting (AAA) to obtain the preshared key for a specific remote peer from an external RADIUS server by configuring `pre-shared-key use-aaa` in an IKE policy. The preshared key can only be obtained using AAA for on-demand IPsec tunnels.

If **`pre-shared-key use-aaa`** is specified, you must configure the preshared key on the AAA server. RADIUS Change of Authorization (CoA) is not supported for dynamically updating preshared keys. Configuring the AAA server is outside the scope of this document.

If **`pre-shared-key use-aaa`** is specified, the AAA server returns the preshared key to use. The format expected by the node is:

```
ike pre-shared-key {hex hex-value|ASCII-value}
```

VSA 203, Security-Service (string) is supported by the SmartEdge router and can appear in Account-Request and Access-Response messages. This VSA specifies an ASE security profile and optionally specifies a preshared key using the following format: **`Security-Service="ike preshared-key hex`** ***`hex-value`*** **`|`** ***`ascii-value`*** **`".`**

8 IPsec Configuration

IPsec configuration involves configuring IPsec proposals (see Section 8.1 on page 32), IPsec policies (see Section 8.2 on page 34), IPsec SAs for manual key IPsec tunnels only (see Section 8.3 on page 36), IPsec ACLs (see Section 8.4 on page 39), and IPsec QoS policies for priority queuing (see Section 8.5 on page 40).

An on-demand auto key IPsec tunnel also requires an IPsec profile whose name must match that of the tunnel. Because of this matching requirement, configuration of the IPsec profile is documented as part of the procedure for configuring an on-demand auto key IPsec tunnel; see Section 9.3.3 on page 51.

8.1 Configuring IPsec Proposals

An IPsec proposal defines the security parameters to be used when establishing an IPsec SA using IKE mode: either or both the AH and ESP protocols, and the encryption and authentication algorithms to use.

A newly configured IPsec proposal is automatically configured with the following default values:

- ESP authentication algorithm: hmac-sha1-96
- ESP encryption algorithm: aes-128-cbc
- No IP compression
- SA lifetime of 86,400 seconds (one day) and no traffic limit

8.1.1 Configuration Tasks

To configure an IPsec proposal:

1. Configure the IPsec proposal in global configuration mode:

   ```
   (config)#ipsec proposal name
   ```

2. In IPsec proposal configuration mode:

   - Configure the description of the IPsec proposal:

     ```
     (config-ipsec-proposal)#description string
     ```

   - Enable (or disable) IP compression:

     ```
     (config-ipsec-proposal)#ip-comp
     ```

     The no form of the command disables IP compression.

   - Configure the AH authentication algorithm:

     ```
     (config-ipsec-proposal)#ah algorithm
     ```

     Note: The **no** form of the command removes the AH configuration.

   - Configure the ESP authentication algorithm:

     ```
     (config-ipsec-proposal)#esp authentication algorithm
     ```

     The **no** form of the command sets the ESP authentication (and ESP encryption) to the default. If either ESP or AH authentication is configured, using the **no** form of the command removes the ESP authentication configuration.

   - Configure the ESP encryption algorithm:

```
(config-ipsec-proposal)#encryption encryption
algorithm
```

If ESP authentication is configured without ESP encryption, the ESP encryption is set to null.

If AH authentication is configured, using the **no** form of the command removes the encryption. If neither ESP authentication or AH is specified, using the **no** form of the command resets the configuration to the default.

- Configure the lifetime for IPsec SAs:

  ```
  (config-ipsec-proposal)#lifetime seconds seconds
  ```

  ```
  (config-ipsec-proposal)#lifetime kbytes kbytes
  ```

3. Commit the transaction.

8.1.2 Configuration Examples

The following example shows how to configure the **ipsec_Prop1** IPsec proposal:

```
[local]Redback(context)#ipsec proposal ipsec_Prop1
[local]Redback(config-ipsec-proposal)#description IPsec-Proposal-1
[local]Redback(config-ipsec-proposal)#ip-comp
[local]Redback(config-ipsec-proposal)#ah hmac-aes-xcbc
[local]Redback(config-ipsec-proposal)#esp authentication hmac-aes-xcbc
[local]Redback(config-ipsec-proposal)#esp encryption aes-256-cbc
[local]Redback(config-ipsec-proposal)#lifetime seconds 43200
[local]Redback(config-ipsec-proposal)#lifetime kbytes 500
```

Example 5 Configuration of an IPsec Proposal

8.2 Configuring IPsec Policies

An IPsec policy provides settings to apply to dynamic-mode IPsec tunnels when establishing connections:

- The size of the antireplay window used to prevent the replay attack and the potential Denial of Service (DoS) attack. A newly created IPsec policy is automatically configured with a default value of 64 packets.
- The DH group used to provide Perfect Forward Secrecy (PFS).
- IPsec proposals used. An IPsec policy can reference up to 16 IPsec proposals.

Figure 7 illustrates the parameters that can be configured for an IPsec policy, and that a policy can include multiple IPsec proposals.

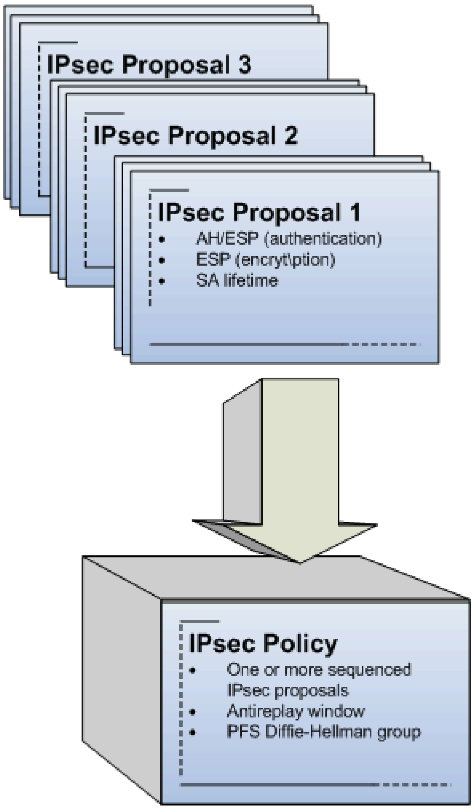

Figure 7 IPsec Policy Parameters

8.2.1 Configuration Tasks

To configure an IPsec policy:

1. Configure the IPsec policy in context configuration mode:

```
(config)#ipsec policy name
```

2. In IPsec policy configuration mode:

   - Configure the description of the IPsec policy:

     ```
     (config-ipsec-policy)#description string
     ```

   - Configure the antireplay window size:

     ```
     (config-ipsec-policy)#anti-replay-window window-size
     ```

   - Enable PFS and specify the DH group used to generate the secret value:

     ```
     (config-ipsec-policy)#perfect-forward-secrecy
     dh-group dh-group
     ```

   - Add up to 16 IPsec proposals, each with a unique sequence number:

     ```
     (config-ipsec-policy)#seq sequence-number proposal
     name
     ```

3. Commit the transaction.

8.2.2 Configuration Examples

The following example shows how to configure the `ipsec_Pol1` IPsec policy:

```
[local]Redback(context)#ipsec policy ipsec_Pol1
[local]Redback(config-ipsec-policy)#description IPsec-Policy-1
[local]Redback(config-ipsec-policy)#anti-replay-window 128
[local]Redback(config-ipsec-policy)#seq 10 proposal ipsec_Prop1
[local]Redback(config-ipsec-policy)#seq 20 proposal ipsec_Prop2
```

Example 6 Configuration of an IPsec Policy

8.3 Configuring IPsec SAs

An IPsec SA is only configured for use by manual key IPsec tunnels. It defines parameters for protecting packets exchanged between each side of the connection of a manual key IPsec tunnel. Each SA is uniquely identified by the Security Parameter Index (SPI), destination IP address, and security protocol used by the connection. The security protocol can be either Authentication Header (AH) or Encapsulating Security Payload (ESP). Each SA is directional, and you can configure inbound and outbound SAs separately or together. When you configure bidirectional SAs, all the attributes of both SAs are identical. When you configure them separately, you must configure them in pairs, using the same security protocol for the inbound SA and outbound SA (although the other attributes can differ).

8.3.1 Configuration Tasks

To configure an IPsec SA:

1. Configure the IPsec SA in global configuration mode:

   ```
   (config)#ipsec security-association sa-name
   ```

2. In IPsec SA configuration mode:

   - Configure the description of the IPsec SA:

     ```
     (config-ipsec-sa)#description string
     ```

   - Configure the antireplay window size:

     ```
     (config-ipsec-sa)#anti-replay-window window-size
     ```

   - Optionally, enable IP compression:

     ```
     (config-ipsec-sa)#ip-comp
     ```

   - Configure the SA attributes for traffic by using the following commands:

     – For bidirectional traffic:

       ```
       (config-ipsec-sa)#both
       ```

     – For inbound traffic:

       ```
       (config-ipsec-sa)#in
       ```

     – For outbound traffic:

       ```
       (config-ipsec-sa)#out
       ```

     Note: The `both` command cannot be used with either the `in` or `out` command.

3. After entering any of these commands, configure the following in IPsec SA SPI configuration mode:

   - If the SA is using the ESP protocol:

     - Specify the ESP SPI value for the inbound traffic, outbound traffic, or bidirectional traffic SAs.

       ```
       (config-ipsec-sa-spi)#esp spi spi-value
       ```

     - Configure the ESP authentication algorithm and the manual key for encrypting inbound, outbound, or bidirectional traffic SAs.

       ```
       (config-ipsec-sa-spi)#esp authentication
       [algorithm] key hex-argument|ASCII-value
       ```

     - Configure one of the following ESP encryption options:

       - The NULL encryption algorithm to indicate that encryption is not required.

         ```
         (config-ipsec-sa-spi)#esp encryption null
         ```

       - An optional encryption algorithm and the manual key for encrypting inbound, outbound, or bidirectional traffic SAs.

         ```
         (config-ipsec-sa-spi)#esp encryption
         [algorithm] key hex-argument|ASCII-value
         ```

   - If the SA is using the AH protocol:

     - Configure the AH SPI value for the inbound traffic, outbound traffic, or bidirectional traffic SAs.

       ```
       (config-ipsec-sa-spi)#ah spi spi-value
       ```

     - Configure an optional AH authentication algorithm and the manual key for authenticating inbound, outbound, or bidirectional traffic SAs:

       ```
       (config-ipsec-sa-spi)#ah [algorithm] key
       hex-argument|ASCII-value
       ```

4. Commit the transaction.

Figure 8 illustrates how separate IPsec policies can be configured for inbound and outbound traffic or a single policy can be configured for bidirectional traffic.

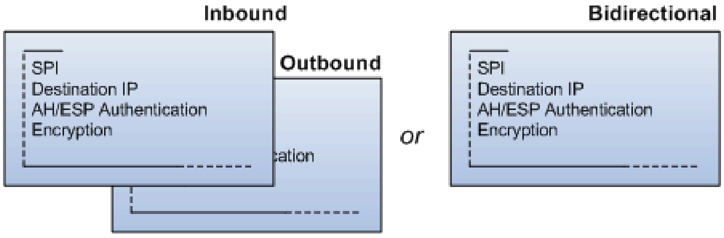

Figure 8 IPsec Policy Configuration

8.3.2 Configuration Examples

The following example shows how to configure the **`ipsec_sa_1`** IPsec SA and the SAs for bidirectional traffic:

```
[local]Redback(context)#ipsec security-association ipsec_sa_1
[local]Redback(config-ipsec-sa-spi)#esp encryption des-cbc key 12345678
[local]Redback(config-ipsec-sa-spi)#esp spi 65535
[local]Redback(config-ipsec-sa-spi)#anti-replay-window 128
[local]Redback(config-ipsec-sa) #both
[local]Redback(config-ipsec-sa-spi)#ah hmac-md5-96 hex 0fa20fa20fa20fa2
[local]Redback(config-ipsec-sa-spi)#ah spi 2546
[local]Redback(config-ipsec-sa-spi)#esp encryption des-cbc key 12345678
[local]Redback(config-ipsec-sa-spi)#esp spi 65535
```

Example 7 Configuration of an IPsec SA

The following example, if applied to the ipsec_sa_1 IPsec SA, shows how to remove all encryption for traffic:

```
[local]Redback(config-ipsec-sa-spi)#esp encryption null
```

8.4 Configuring IPsec ACLs

Use IPsec ACLs to identify the traffic to be secured in the IPSec tunnel. An IPsec ACL consists of an ordered list of rules, each of which defines a class of packets, that defines a traffic filter. To identify the traffic to be secured, an IPsec ACL rule specifies the source address, the source port, the protocol, the destination address, and the destination port. Wildcards, such as masks, can be used to broaden the scope of the filter. When a manual-mode IPsec tunnel is configured, you can associate an IPsec ACL with an IPsec SA to filter the traffic; however, only the first rule in the IPSec ACL is used to filter traffic. When a dynamic-mode IPsec tunnel is configured, you can associate an IPsec ACL with an IPsec policy to filter the traffic. In either case, if no IPsec ACL is specified, all traffic routed to the tunnel is matched.

8.4.1 Configuration Tasks

To configure an IPsec policy:

1. Configure the IPsec ACL in context configuration mode:

   (config-ctx)#**ipsec access-list** ***acl-name***

2. In IPsec ACL configuration mode:

   - Configure the description of the IPsec ACL:

     (config-ipsec-acl)#**description** ***string***

   - Create IPsec ACL statements to identify packets that meet the specified criteria:

     (config-ipsec-acl)#**seq** ***sequence-number*** [**protocol**] [***source-network-prefix/source-prefix-length***|**any**] {**eq** ***source-port***} [***dest-network-prefix/dest-prefix-length***|**any**] [**cond** ***dest-port***]

3. Commit the transaction.

8.4.2 Configuration Examples

The following example shows how to configure the **ipsec_ACL1** IPsec ACL with three rules:

```
[local]Redback(config-ctx)#ipsec access-list ipsec_ACL1
[local]Redback(config-ipsec-acl)#description IPsec-ACL-1
[local]Redback(config-ipsec-acl)#seq 10 tcp 1.1.1.0/24 eq 20000
[local]Redback(config-ipsec-acl)#seq 20 1.1.1.0/24 2.2.2.0/24
[local]Redback(config-ipsec-acl)#seq 30 any any
```

Example 8 Configuration of an IPsec ACL

8.5 Configuring IPsec QoS Policies for Priority Queuing

An IPsec QoS policy for priority queuing specifies whether four queues with strict priority scheduling are enabled on each SA when an IPsec tunnel is set up. Priority queuing reduces the impact of latency and delay when oversubscription occurs. You can configure an IPsec QoS policy for priority queuing with one queue (no priority queuing) or four queues (priority queuing).

When an IPsec tunnel references an IPsec QoS policy for priority queuing, the Priority Descriptor (PD) or Differentiated Services Code Point (DSCP) bits in data packets are used to determine the priority of the packet and direct it to the appropriate queue. If QoS remarking is configured on the ingress line card, the remarked values that override the PD values are used.

Table 3 lists how the DSCP or PD values in a packet are directed to the four queues when they are enabled on an IPsec SA. The higher the DSCP/PD value the higher the priority of the packet. The lower the value of the SA queue the higher the priority of the queue.

Table 3 DSCP/DP Value to IPsec SA Priority Queue Mapping

DSCP/PD Value	SA Queue
CS0	3
CS1	
CS2	2
CS3	
CS4	1
CS5	
CS6	0

If no IPsec QoS policy for priority queuing is specified for an IPsec tunnel, only one queue is used and all traffic is treated as though it has the same priority.

8.5.1 Configuration Tasks

To configure an IPsec QoS policy for priority queuing:

1. Create or modify the IPsec QoS policy in global configuration mode:

   ```
   #(config)ipsec qos policy name pq
   ```

2. Specify the number of priority queues in IPsec Policy priority queue configuration mode:

   ```
   (config-ipsec-policy-pq)#num-queues {1|4}
   ```

8.5.2 Configuration Examples

The following example shows how to configure the **`ipsec-qos-pq-ipsecsa1`** IPsec QoS policy for priority queuing:

```
[local]Redback(config)#ipsec qos policy ipsec
-qos-pq-ipsecsa1 pq
[local]Redback(config-ipsec-policy-pq)#num-queues 4
```

Example 9 Configuration of an IPsec QoS Policy

9 IPsec Tunnel Configuration

You can create an IPsec tunnel endpoint in circuit-based mode or in economical mode, regardless of how the SAs are negotiated. An economical mode IPsec tunnel has no presence on any line card and requires no resources on the line card, allowing more tunnels to be created per line card; however, you must use the default circuit-based mode for IPsec tunnels that use dynamic routing protocols over the tunnel.

By default, an IPsec tunnel is created in circuit-based mode; you must explicitly specify the economical mode.

The IPsec SAs used to secure the connection for each tunnel endpoint can be negotiated automatically using preshared keys, or configured manually using preconfigured keys. IPsec SAs configured manually use preconfigured keys and do not scale well, are more prone to configuration errors, and require additional maintenance. IPsec SAs negotiated automatically are more secure and require less maintenance. Automatic negotiation of IPsec SAs uses either the IKEv1 or IKEv2 protocol. The IKEv2 protocol is efficient, allows each endpoint to enforce its own lifetime policy to support rekeying, supports more robust encryption, allows each end point to use different methods to authenticate peers, and allows traffic selector values to be negotiated.

Three types of tunnels can be configured:

- Static auto key IPsec tunnels connect a known local endpoint to a remote endpoint whose IP address is known, using IPsec SAs negotiated with the IKE protocol.

- On-demand auto key IPsec tunnels connect a known local endpoint to a remote endpoint whose IP address is unknown at configuration, using IPsec SAs negotiated with the IKE protocol.

- Manual key IPsec tunnels connect a known local endpoint to a known remote endpoint, using IPsec SAs manually configured at each endpoint.

Optionally, you can use traffic-selector guided route addition in a static or on-demand auto key IPsec tunnel configuration. Traffic-selector guided route addition uses the traffic selectors negotiated as part of the IKE negotiations used to establish the IPsec SAs to add routes from the local endpoint to the network protected by the remote peer. When enabled, traffic-selector guided route addition automatically adds and deletes IP routes dynamically to point to the IPsec tunnel as the tunnel comes up or goes down. Otherwise, you must you must explicitly add static IP routes to point to the IPsec tunnel or run dynamic routing protocols over the tunnel.

See:

- Section 9.2 on page 46 for configuration of auto key IPsec tunnels.

- Section 9.3 on page 49 for configuration of on-demand auto key tunnels.
- Section 9.4 on page 53 for configuration of manual key tunnels.

Figure 9 illustrates the components of an IPsec tunnel endpoint configuration.

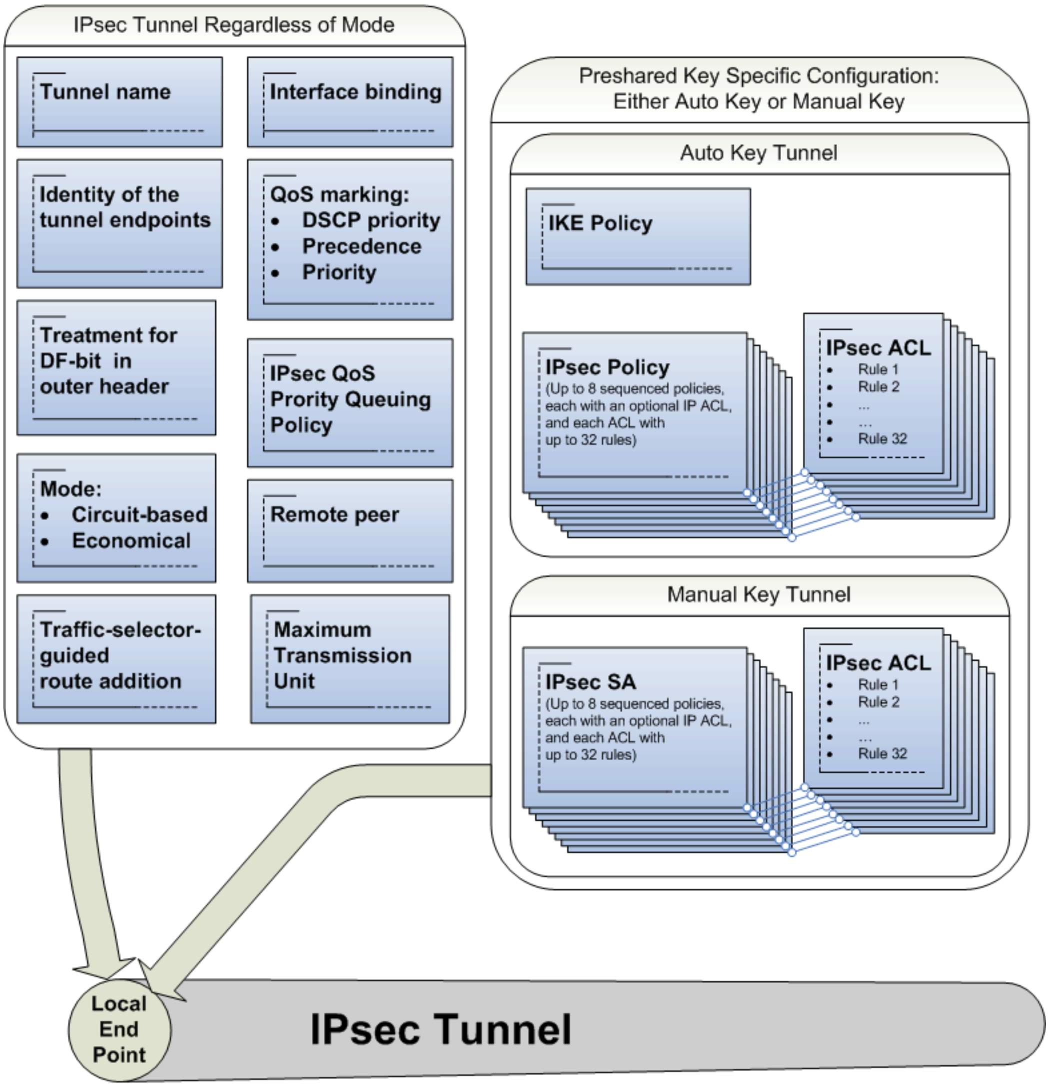
IPsec Tunnel Regardless of Mode
Tunnel name
Interface binding
Identity of the tunnel endpoints
QoS marking:
• DSCP priority
• Precedence
• Priority
Treatment for DF-bit in outer header
IPsec QoS Prority Queuing Policy
Mode:
• Circuit-based
• Economical
Remote peer
Traffic-selector-guided route addition
Maximum Transmission Unit
Preshared Key Specific Configuration: Either Auto Key or Manual Key
Auto Key Tunnel
IKE Policy
IPsec Policy
(Up to 8 sequenced policies, each with an optional IP ACL, and each ACL with up to 32 rules)
IPsec ACL
• Rule 1
• Rule 2
• ...
• ...
• Rule 32
Manual Key Tunnel
IPsec SA
(Up to 8 sequenced policies, each with an optional IP ACL, and each ACL with up to 32 rules)
IPsec ACL
• Rule 1
• Rule 2
• ...
• ...
• Rule 32
Local End Point
IPsec Tunnel

Figure 9   Components of an IPsec Tunnel Endpoint Configuration

9.1 Prerequisites

In addition to the context-level prerequisites listed in Section 3 on page 7, the following items must be configured before they can be specified in an IPsec tunnel:

- For an auto key or on-demand auto key IPsec tunnel:
  - IKE proposal, see Section 6.2 on page 21
  - IKE policy, see Section 6.4 on page 25
  - IPsec proposal, see Section 8.1 on page 32
  - IPsec policy, see Section 8.2 on page 34
  - IPsec ACLs, see Section 8.4 on page 39
  - IPsec QoS policy (optional—if not used only a single queue is used), see Section 8.5 on page 40
- For a manual key IPsec tunnel:
  - IPsec SAs, see Section 8.3 on page 36
  - IPsec ACLs, see Section 8.4 on page 39
  - IPsec QoS policy (optional—if not used only a single queue is used), see Section 8.5 on page 40

9.2 Configuring a Static Auto Key IPsec Tunnel Endpoint

A static auto key tunnel can only be configured between a known local endpoint and a known remote endpoint and uses the IKE protocol to automatically negotiate IPsec SAs. A static auto key tunnel supports tunnel-interface-connected, static, and traffic-selector-based routes in both circuit-based and economical modes. Additionally, if the circuit-based mode is used, OSPF, RIP, and IGPs dynamic routes are supported, and if economical mode is used, only BGP dynamic routes are supported. You need to configure individual tunnels to connect the same local endpoint to multiple remote endpoints.

For a static auto key IPsec VPN endpoint, you must configure:

- The static binding between the tunnel and the interface configured for this IPsec tunnel.
- The remote peer.
- The identity of the tunnel endpoints.
- An IKE policy to negotiate the connection with the remote peer.
- Optionally, an IPsec QoS policy for priority queuing to instantiate priority queues for each SA established for the tunnel.
- Up to eight sequenced IPsec policies, each optionally with an IPsec ACL, to use when establishing connections with the remote peer.
- Optionally, the setting for how the Don't Fragment (DF) bit for the outer header is treated. By default, the DF bit setting used in the inner IP heading is copied to the outer IP heading.

9.2.1 Configuration Tasks

To configure the endpoints of an auto key IPsec tunnel, do the following:

1. Create an IPsec tunnel endpoint:

   - Circuit-based mode, which is required for IPsec tunnels that use the OSPF or RIP dynamic routing protocols:

     ```
     (config)#tunnel ipsec name
     ```

   - Economical mode, which makes more efficient use of line cards:

     ```
     (config)#tunnel ipsec name economical
     ```

2. Configure the static bind between the tunnel endpoint and the interface configured for this IPsec tunnel endpoint.

   ```
   (config-tunnel)#bind interface if-name [context-name]
   ```

3. Specify how the DF bit for the outer IP header should be treated.

   ```
   (config-tunnel)#df-bit {propagate|set|clear}
   ```

4. Specify the IKE policy used to negotiate the connection with the remote peer:

   ```
   (config-tunnel)#ike-policy ike-policy-name
   ```

5. Optionally, specify an IPsec QoS policy for priority queuing.

   ```
   (config-tunnel)#qos policy queuingipsec-qos-policy-pq-name
   ```

6. Optionally, enable traffic-selector guided route addition :

   ```
   (config-tunnel)#ip route traffic-selector-guided
   ```

7. Specify the tunnel endpoints:

   ```
   (config-tunnel)#peer-end-point local loc-ip-addr remote
   rem-ip-addr [context ctx-name]
   ```

   Use the IP address of the loopback interface defined to identify the local gateway for IPsec tunnels configured on this SmartEdge router as the value for the **local** ***loc-ip-addr*** construct. Similarly, use the IP address configured on the remote device as the value for the **remote** ***rem-ip-addr*** construct. This allows you to create multiple IPsec tunnels with the same local endpoint, each with a different remote endpoint that can be part of the same IPsec VPN. Identify the context that contains the interface to the local end of the tunnel, if it is not in context `local`.

8. Identify the remote peer to use when negotiating IKE requests for the remote endpoint:

   ```
   (config-tunnel)#remote-id remote_id
   ```

9. Optionally, specify the Maximum Transmission Unit (MTU) of packets sent in IPsec tunnel.

   ```
   (config-tunnel)#mtu size
   ```

10. Assign up to eight sequenced IPsec policies, each optionally with an IPsec ACL:

    ```
    (config-tunnel)#seq id ipsec-policy ipsec-pol-name
    [access-group ipsec-acl-name]
    ```

    If no access group is specified, an implicit wildcard policy that matches all traffic is used.

9.2.2 Configuration Examples

The following example shows how to create a static auto key IPsec tunnel, **`IPsec_tunnel_1`**, using IKE negotiations to establish the SAs between the two endpoints. The IPsec tunnel includes two IPsec policies, each with an associated IPsec ACL:

```
[local]Redback(config)#tunnel ipsec IPsec_tunnel_1
[local]Redback(config-tunnel)#ike policy IKE_Pol1
[local]Redback(config-tunnel)#df-bit clear
[local]Redback(config-tunnel)#bind interface ipsec-if1
[local]Redback(config-tunnel)#qos policy queuing ipsec-qos-pq-3
[local]Redback(config-tunnel)#ip route traffic-selector-guided
[local]Redback(config-tunnel)#peer-end-point local 172.16.1.1 remote 172.16.1.2
[local]Redback(config-tunnel)#remote-id 72.0.0.1
[local]Redback(config-tunnel)#mtu 256
[local]Redback(config-tunnel)#seq 10 ipsec-policy ipsec_Pol1 access-group ipsec_ACL1
[local]Redback(config-tunnel)#seq 20 ipsec-policy ipsec_Pol2 access-group ipsec_ACL2
```

Example 10 Configuration of a Static Auto Key IPsec Tunnel

9.3 Configuring an On-Demand Auto Key IPsec Tunnel

An on-demand auto key tunnel can be configured between a known local endpoint and an unknown remote endpoint and uses the IKE protocol to automatically negotiate IPsec SAs. If the IKEv1 protocol is used to negotiate IPsec SAs, only the aggressive mode for key exchange is supported. An on-demand auto key tunnel supports traffic-selector-based routes and, if circuit-based mode is used, OSPF and RIP dynamic routes. Tunnel interface and static routes are not supported. A single on-demand tunnel configuration using the same local IP address and local ID can bring up multiple remote endpoints dynamically.

During IPsec tunnel configuration, the IP address of the remote peer may not be known. For example, remote CPE devices that do not have a static IP address obtain it dynamically at bootup. In this situation, you can configure an on-demand auto key IPsec tunnel.

For on-demand IPsec tunnels, the remote peer is identified by its IKE identity; only one on-demand IPsec tunnel can be active for each remote IKE identity. The remote IKE identity can be a fully-specified ID for IP address and FQDN formats, a wildcard for IP addresses, or a partial ID for FQDN in the format `*.xxx.xxx`, or `*.*.xxx`. If remote IKE identity configurations for a tunnel overlap, the order of matching is:

1. Exact ID match
2. Partial ID match
3. Wildcard match

The same remote IKE ID should not be used in more than one IKE policy; however, the same remote IKE ID can be used in configuring multiple on-demand IPsec tunnels and IKE policies.

The following routing protocols are supported over IPsec tunnels bound to IPsec multibind interfaces:

- OSPFv2
- RIPv2

For OSPF, each of the IPSec tunnels is treated as the Point-to-Multipoint (P2MP) unnumbered interface. All OSPF control packets are sent using the multicast address. If a routing protocol is enabled over an IPsec multibind interface, then all tunnels bound to a multibind interface run the same routing protocol. You must not exceed the maximum number of OSPF adjacencies.

Equal-Cost Multipath (ECMP) across IPsec tunnels is not supported.

The same context-id and remote-id cannot be used in both static and on-demand auto key tunnel configuration. All IPsec tunnels for remote unknown peers that use the same context-id and local-ip for IKE signaling are mapped to the same ASP.

Tunnels that use different local IP addresses for a given context are load-balanced across the different ASPs of the ASP group that the context is associated with. If the number of tunnels per context exceeds the capacity of a single ASP, multiple local IP addresses must be used to achieve load-balancing across multiple ASPs.

If you configure **`pre-shared-key use-aaa`**, in the IKE policy referenced by the on-demand tunnel, the context associated with on-demand tunnel interface is used for AAA authorization to obtain a preshared key from the RADIUS server. If the RADIUS server specified in the context is unreachable, and a fallback to global authentication is configured in that context, then the RADIUS server configured in the local context is used.

If you do not configure **`pre-shared-key use-aaa`**, or if RADIUS does not specify an IPsec profile name, the system attempts to find an IPsec profile with the same name as the on-demand IPsec tunnel. If no matching tunnel is found, an error occurs and the tunnel bringup fails.

For an on-demand auto key IPsec VPN endpoint, you must configure:

- A multibind IPsec interface
- An IPsec tunnel
- An IPsec profile to specify how traffic in the IPsec tunnel should be handled

9.3.1 Configuring an Interface for an On-demand IPsec Tunnel

In addition to completing the context-level prerequisites for IPsec tunnel configuration as specified in Section 3 on page 7, you must enable an interface to have multiple IPsec tunnel circuits bound to it. This IPsec multibind interface must be unnumbered to enable IP processing on the interface without assigning it an explicit IP address.

```
(config-ctx)##interface name ipsec multibind
```

```
config-if)#ip unnumbered if-name
```

The IPsec multibind interface can be configured in any context.

9.3.2 Creating an On-demand IPsec Tunnel

To create an IPsec tunnel endpoint:

1. Create the endpoint in one of the supported modes:

    - Circuit-based mode, which is required for IPsec tunnels that use the OSPF or RIP dynamic routing protocols:

```
(config)#tunnel ipsec name on-demand
```

- Economical mode, which makes more efficient use of line cards:

  ```
  (config)#tunnel ipsec name on-demand economical
  ```

2. Specify the tunnel endpoint:

   ```
   (config-tunnel)#peer-end-point local loc-ip-addr
   [context ctx-name]
   ```

   Use the IP address of the loopback interface defined to identify the local endpoint for IPsec tunnels configured on this SmartEdge router as the value for the **local** *loc-ip-addr* construct. Using the IP address of a loopback interface allows the local IP address to remain up, providing a stable environment for routing. Identify the context that contains the interface to the local end of the tunnel, if it is not in context `local`.

3. Specify the IKE policy used to negotiate the connection with the remote peer:

   ```
   (config-tunnel)#ike-policy ike-policy-name
   ```

4. Specify the maximum number of tunnels per profile in this on-demand tunnel.

   ```
   (config-tunnel)#max-tunnels value
   ```

5. Configure the binding between the on-demand tunnel and the IPsec multibind interface configured for this on-demand IPsec tunnel in Section 9.3.1 on page 50.

   ```
   (config-tunnel)#bind interface if-name [context-name]
   ```

6. Optionally, specify an IPsec QoS policy for priority queuing.

   ```
   (config-tunnel)#qos policy queuing ipsec-qos-policy-pq-n
   ame
   ```

7. Optionally, enable traffic-selector guided route addition :

   ```
   (config-tunnel)#ip route traffic-selector-guided
   ```

9.3.3 Creating an IPsec Profile

Configure the IPsec profile to specify how traffic in the on-demand IPsec tunnel should be handled. The IPsec profile must be created in the same context as the multibind interface configured in Section 9.3.1 on page 50.

To create an IPsec profile:

1. In context configuration mode, enter the following command:

```
(config-ctx)#ipsec profile profile-name
```

The name of the IPsec profile should match the name of the on-demand IPsec tunnel.

2. Specify how the DF bit for the IP header should be treated.

   ```
   (cfg-ipsec-profile)#df-bit {set | clear}
   ```

3. Specify the MTU of packets sent in IPsec tunnel.

   ```
   (cfg-ipsec-profile)#mtu size
   ```

4. Assign up to eight sequenced IPsec policies, each optionally with an IPsec ACL:

   ```
   (cfg-ipsec-profile)#seq id ipsec-policy ipsec-pol-name
   [access-group ipsec-acl-name]
   ```

   If an access group is specified, it must exist in the same context as the IPsec profile. If no access group is specified, an implicit wildcard policy that matches all traffic is used.

9.3.4 Configuration Examples

The following example shows how to create an on-demand IPsec tunnel, **`IPsec_tunnel_2`**, using IKE negotiations to establish the SAs between the two endpoints. The IPsec tunnel is bound to the IPsec multibind interface `ipsec_mb_se_1`, which uses the IP address from interface `peer_local_ip_1`. The IPsec tunnel includes an IPsec profile with the name **`IPsec_tunnel_2`** matching the IPsec tunnel name, and including two IPsec policies, each with an associated IPsec ACL:

```
[local]Redback(config)#context isp1
[local]Redback(config-ctx)#interface ipsec_mb_se_1 ipsec multibind
[local]Redback(config-if)#ip unnumbered peer_local_ip_1
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#exit
[local]Redback(config)#tunnel ipsec se_1 on-demand
[local]Redback(config-tunnel)#peer-end-point local 39.0.0.1 context isp1
[local]Redback(config-tunnel)#ike-policy ike_pol_se_1
[local]Redback(config-tunnel)#bind interface ipsec_mb_se_1 isp1
[local]Redback(config-tunnel)#max-tunnels 50
[local]Redback(config-tunnel)#qos policy queuing ipsec-qos-pq-3
[local]Redback(config-tunnel)#ip route traffic-selector-guided
[local]Redback(config-tunnel)#exit
[local]Redback(config)#context isp1
[local]Redback(config-ctx)#ipsec profile IPsec_tunnel_2
[local]Redback(cfg-ipsec-profile)#seq 1 ipsec-policy ipsec_pol access-group ipsec_ACL1
[local]Redback(config-tunnel)#seq 20 ipsec-policy ipsec_Pol2 access-group ipsec_ACL2
```

Example 11 Configuration of an On-Demand IPsec Tunnel

9.4 Configuring a Manual Key IPsec Tunnel

A manual key IPsec tunnel can only be configured between a known local endpoint and a known remote endpoint and requires that compatible IPsec SAs are manually configured at both endpoints. A manual key tunnel supports tunnel-interface- connected and static mode routes in both circuit-based and economical modes. Traffic-selector-based routes are not supported. Additionally, if circuit-based mode is used, OSPF, RIP, and IGPs dynamic routes are supported, and if economical mode is used, only BGP dynamic routes are supported. Manual key tunnel configuration is not easily scalable and is unsuitable if more than a small number of tunnels is required.

For a manual key IPsec VPN endpoint, you must configure:

- The static binding between the tunnel and the interface configured for this IPsec tunnel
- The remote peer
- The identity of the tunnel endpoints
- Up to eight sequenced manual-keyed SAs
- Optionally, the setting for how the Don't Fragment (DF) bit for the outer header is treated. By default, DF bit setting used in the inner IP heading is copied to the outer IP heading.

9.4.1 Configuration Tasks

To configure the endpoints of a manual key IPsec tunnel:

1. Create an IPsec tunnel endpoint:

   - Circuit-based mode, which is required for IPsec tunnels that use the OSPF or RIP dynamic routing protocols:

     ```
     (config)#tunnel ipsec name manual
     ```

   - Economical mode, which makes more efficient use of line cards:

     ```
     (config)#tunnel ipsec name manual economical
     ```

2. Configure the static bind between the tunnel endpoint and the interface configured for this IPsec tunnel endpoint.

   ```
   (config-tunnel)#bind interface if-name [context-name]
   ```

3. Specify how the DF bit for the outer IP header should be treated.

   ```
   (config-tunnel)#df-bit {propagate | set | clear}
   ```

4. Specify the identity of the tunnel endpoints:

```
(config-tunnel)#peer-end-point local loc-ip-addr remote
rem-ip-addr [context ctx-name]
```

Use the IP address of the loopback interface defined to identify the local gateway for IPsec tunnels on this SmartEdge router as the value for the `local` *`loc-ip-addr`* construct. Similarly, use the IP address configured on the remote device as the value for the `remote` *`rem-ip-addr`* construct. This allows you to create multiple IPsec tunnels with the same local endpoint, each with a different remote endpoint that can be part of the same IPsec VPN.

5. Optionally, specify an IPsec QoS policy for priority queuing.

```
(config-tunnel)#qos policy queuing ipsec-qos-policy-pq-name
```

6. Optionally, specify the MTU of packets sent in IPsec tunnel.

```
(config-tunnel)#mtu size
```

7. Assign up to eight sequenced manual-keyed SAs, each optionally with an IPsec ACL:

```
(config-tunnel)#seq id ipsec-policy name security-association sa-name [access-group ipsec-acl-name]
```

If no access group is specified, an implicit wildcard policy that matches all traffic is used. For each manual SA with an associated ACL, only the traffic pattern identified in the first rule of the ACL is matched.

9.4.2 Configuration Examples

The following example shows how to create the manual key **`manualIPsec_tunnel_1`** IPsec tunnel. The IPsec tunnel includes two manually-keyed SAs, each with an associated IPsec ACL:

```
[local]Redback(config)#tunnel ipsec manual_IPsec_tunnel_1 manual
[local]Redback(config-tunnel)#df-bit clear
[local]Redback(config-tunnel)#bind interface ipsec-if1
[local]Redback(config-tunnel)#qos policy queuing ipsec-qos-pq-3
[local]Redback(config-tunnel)#peer-end-point local 172.16.1.1 remote 172.16.1.2
[local]Redback(config-tunnel)#remote-id 72.0.0.1
[local]Redback(config-tunnel)#mtu 256
[local]Redback(config-tunnel)#seq 10 security association ipsec_sa_1 access-group ipsec_ACL1
[local]Redback(config-tunnel)#seq 20 security association ipsec_sa_2 access-group ipsec_ACL2
```

Example 12 Configuration of a Manual Key IPsec Tunnel

10 IPsec Monitoring Commands

Show commands display a variety of information for IPsec tunnel endpoints, as shown in Table 4. Enter show commands in any mode.

Table 4 IPsec Monitoring Commands

To display the following information...	Enter this command...
IKE configuration in the current context or all contexts	**show configuration ike** [**all-contexts**] [**verbose**]
IPsec configuration in the current context or all contexts	**show configuration ipsec** [**all-contexts**] [**verbose**]
Tunnel configuration in the current context or all contexts	**show configuration tunnel** [**all-contexts**] [**verbose**]
One or more IKE policies configured in the IPsec VPN application for the specified ASP; context-specific	**show ike** [**card** *slot-id/asp-id*] **policy** [*policy-name*]
One or more IKE proposals configured in the IPsec VPN application for the specified ASP	**show ike**[**card** *slot-id/asp-id*] **proposal** [*proposal-name*]
Show IKEv1 or IKEv2 statistics for the specified ASP	**show ike card** *slot-id/asp-id* **statistics global**{**ike1**|**ike2**}
Show IPsec statistics for the specified ASP and context	**show ipsec card** *slot/asp-id* **statistics global context** *context-name*
Show IKE statistics per tunnel	**show tunnel ipsec name** *tunnel-name* **statistics ike** [**detail**]
One or more IPsec ACLs configured in the IPsec VPN application for the specified ASP; context-specific	**show ipsec** [**card** *slot-id/asp-id*] **access-list** [*ipsec-acl-name*]
One or more IPsec policies configured in the IPsec VPN application for the specified ASP	**show ipsec**[**card** *slot-d/asp-id*] **policy** [*policy-name*]
One or more IPsec profiles configured in the IPsec VPN application for the specified ASP	**show ipsec** [**card** *slot-d/asp-id*] **profile** [*profile-name*]
One or more IPsec proposals configured in the IPsec VPN application for the specified ASP	**show ipsec** [**card** *slot/asp-id*] **proposal** [*proposal-name*]

To display the following information...	Enter this command...
One or more IPsec SAs configured in the IPsec VPN application for the specified ASP	**show ipsec** [**card** *slot*/*asp-id*] **security-association** [*sa-name*]
One or more IPsec tunnels configured in the IPsec VPN application	**show tunnel ipsec** [[**name** *tunnel-name* | **remote** *ip-address*] [**detail**]] | [[**name** *tunnel-name*] **on-demand**]
Statistics associated with the specified IPsec tunnel	**show tunnel ipsec** *tunnel-name* [**on-demand**] **statistics** [**Detail**]
Information about trusted or self certificates	**show pki** [**asp** *slot-id*/*asp-id*] **certificate** {**trusted** | **self**} **rsa** [**identity** *identity* |**handle** *handle*
Information about the public-private key pairs, excluding the actual keys	**show pki key-pair** [*key-pair-name*]
Information about all certificate requests and their associated file names; context specific.	**show pki certificate-request rsa** [**all**]

For more information about **show** commands see Reference [7].

11 IPsec VPN SA Clearing Commands

Without clearing SAs, if you make any configuration changes to IPsec tunnels (except changes to IPsec ACLs) that affect SAs, the changes do not take effect until the current SA lifetime expires and new SAs are negotiated and established. Clear SAs when you want an updated IPsec configuration setup to take effect immediately after making the changes. These commands restart the SAs with the most current configuration settings. If the IPsec tunnel is in manual mode, new manual SAs are created immediately. If IPsec tunnel SAs are created using the IKE protocol, the SAs and attributes are renegotiated using IKE before new ones are established.

You can clear the SAs associated with specific policies and IPsec tunnels:

- To clear IKE SAs associated with a specific tunnel, enter the following command in exec mode:

  ```
  clear ike sa tunnel tunnel-name
  ```

- To clear IPsec SAs associated with a specific tunnel, enter the following command in exec mode:

  ```
  clear ipsec sa tunnel tunnel-name
  ```

For more information about `clear` commands see Reference [7].

12 Sample End-to-End Configurations

Example 13, Example 14, and Example 15 are excerpts from SmartEdge router configuration files. The first two examples provide sample configurations from both SmartEdge routers at each end of the same IPsec tunnel to show a complete tunnel configuration; these examples show how IKEv1 is configured to negotiate the SAs used for the tunnel. The third example is an excerpt from the configuration file for one SmartEdge router at one end of an IPsec tunnel that shows how IKEv2 is configured to negotiate the SAs.

```
!
asp pool pool1 service security
 asp 1/1
asp pool pool2 service security
 asp 1/2
asp group group1
 pool pool1
 asp-count 1
asp group group2
 pool pool2
 asp-count 1
!
context local
!
 no ip domain-lookup
!
 interface mgmt
  ip address 10.192.16.250/23
 logging console
!
!
 administrator test encrypted 1
     $1$........$kvQfdsjs0ACFMeDHQ7n/o.
   privilege start 15
   no timeout session idle
!
!
 ip route 0.0.0.0/0 10.192.17.254
!


!
context vpnEndA
!
 no ip domain-lookup
!
 interface N2X-port1
  ip address 50.0.0.254/8
```

```
!
 interface ipsec_local_loopback1 loopback
  ip address 30.0.0.1/32
 !
 interface to_ipsec_peer
  ip address 40.0.0.1/16
!
 interface tunnel_ipsec_1
  ip address 55.0.1.1/30
 no logging console
!
 ip route 20.0.0.0/8 40.0.0.2
 ip route 60.1.0.1/32 tunnel_ipsec_1
 !
!
 asp-group group1 service security
 ike policy ike-pol1
  mode aggressive
  identity local 30.0.0.1
  preshared-key 123456789123456
  identity remote 20.1.0.2
  seq 10 proposal ike-prop
!
ipsec access-list any-acl
 seq 10 any any
!
!
! ** End Context **
 logging tdm console
 logging active
 logging standby short
!
!
tunnel ipsec rec_1_1
 peer-end-point local 30.0.0.1 remote 20.1.0.2 context
   vpnEndA
 bind interface tunnel_ipsec_1 vpn1
 remote-id 20.1.0.2
 ike-policy ike-pol1
 seq 10 ipsec-policy ipsec-pol access-group any-acl
!
!
!Ethernet connectivity fault management configuration
!
ike proposal ike-prop
 authentication algorithm hmac-md5-96
 encryption algorithm des-cbc
 dh-group 1
 !
ipsec proposal ipsec-prop
 esp encryption 3des-cbc
 esp authentication hmac-sha1-96
 !
ipsec policy ipsec-pol
 seq 10 proposal ipsec-prop
!
!
card ase 1
!
card ge-10-port 3
!
port ethernet 3/1
 no shutdown
 bind interface N2X-port1 vpnEndA
!
port ethernet 3/11
 no shutdown
 bind interface to_ipsec_peer vpnEndA
!
!
port ethernet 7/1
! XCRP management ports on slot 7 and 8 are
! configured through 7/1
 no shutdown
 bind interface mgmt local
!
!
 boot configuration ipsec-demo-15.cfg
!
no service console-break
!
service crash-dump-dram
!
no service auto-system-recovery
!
end
```

Example 13 Configuration of the SmartEdge Router at End A of an IPsec Tunnel

```
!
asp pool pool1 service security
 asp 1/1
asp pool pool2 service security
 asp 1/2
asp group group1
 pool pool1
 asp-count 1
asp group group2
 pool pool2
 asp-count 1
!
context local
```

```
!
 no ip domain-lookup
!
 interface mgmt
  ip address 10.192.16.250/23
 logging console
!
!
 administrator test encrypted 1
       $1$........$kvQfdsjs0ACFMeDHQ7n/o.
   privilege start 15
   no timeout session idle
!
!
 ip route 0.0.0.0/0 10.192.17.254
!


!
context vpnEndB
!
 no ip domain-lookup
!
 interface N2X-port2
  ip address 60.0.0.254/8
!
 interface ipsec_local_loopback1 loopback
  ip address 20.1.0.2/32
!
 interface to_ipsec_peer
  ip address 40.0.0.2/30

 interface tunnel_ipsec_1
  ip address 55.0.1.2/30
!
 no logging console
!
 ip route 30.0.0.0/8 40.0.0.1
 ip route 50.1.0.1/32 tunnel_ipsec_1
!


!
 asp-group group2 service security
 ike policy ike-pol1
  mode main
  identity local 20.1.0.2
  preshared-key 123456789123456
  identity remote 30.0.0.1
  seq 10 proposal ike-prop
```

```
!
ipsec access-list any-acl
 seq 10 any any
!
!
! ** End Context **
 logging tdm console
 logging active
 logging standby short
!
!
tunnel ipsec rec_2_1
 peer-end-point local 20.1.0.2 remote 30.0.0.1 context
   vpnEndB
 bind interface tunnel_ipsec_1 vpn2
 remote-id 30.0.0.1
 ike-policy ike-pol1
 seq 10 ipsec-policy ipsec-pol access-group any-acl
!
!
!Ethernet connectivity fault management configuration
!
ike proposal ike-prop
 authentication algorithm hmac-md5-96
 encryption algorithm des-cbc
 dh-group 1
 !
ipsec proposal ipsec-prop
 esp encryption 3des-cbc
 esp authentication hmac-sha1-96
 !
ipsec policy ipsec-pol
 seq 10 proposal ipsec-prop
!
!
card ase 1
!
card ether-12-port 3
!
port ethernet 3/2
 no shutdown
 bind interface N2X-port2 vpnEndB
!
port ethernet 3/12
 no shutdown
 bind interface to_ipsec_peer vpnEndB
!
!
port ethernet 7/1
! XCRP management ports on slot 7 and 8
! are configured through 7/1
```

```
 no shutdown
 bind interface mgmt local
!
!
 boot configuration ipsec-demo-15.cfg
!
no service console-break
!
service crash-dump-dram
!
no service auto-system-recovery
!
end
```

Example 14 Configuration of the SmartEdge Router at End B of an IPsec Tunnel

```
!
config
!
asp pool ipsec_pool service security
 asp 4/1
asp group ipsec_group
 pool ipsec_pool
 asp-count 1
!
!
! IKE2 Proposal Configuration
!
ike2 proposal ike_proposal
 authentication algorithm hmac-md5-96
 encryption algorithm 3des-cbc
 pseudo-random-function hmac-sha1
 dh-group 2
!
ipsec proposal ipsec_proposal
 esp encryption 3des-cbc
 esp authentication hmac-md5-96
 ah hmac-sha1-96
 lifetime seconds 1800
!
ipsec policy ipsec_policy
 perfect-forward-secrecy dh-group 2
 seq 1 proposal ipsec_proposal
!
context vpn
!
 interface ipsec ipsec
  ip address 90.0.0.1/8
!
 interface ipsec_loop loopback
  ip address 30.0.0.1/32
```

```
!
 interface peer_interface
  ip address 20.0.0.1/8
!
 asp-group ipsec_group service security
!
 ike2 policy ike_policy  IKE2 POLICY
  connection-type both
  identity local 30.0.0.1
  pre-shared-key 1234567890123456
  identity remote 20.0.0.2
  seq 1 proposal ike_proposal
!
ipsec access-list ipsec_acl
 seq 1 any any
!
tunnel ipsec ipsec_tunnel
 peer-end-point local 30.0.0.1 remote 20.0.0.2 context vpn
 bind interface ipsec vpn
 remote-id 20.0.0.2
 ike-policy ike_policy
 seq 1 ipsec-policy ipsec_policy access-group ipsec_acl
!
```

Example 15   Sample IKEv2 Configuration

13 High Availability Configuration

You can provide high availability for IPsec tunnels by setting up redundant connections between the local SmartEdge router and the remote security gateway. You configure two tunnels in different security-enabled contexts and configure routing to switch between tunnels. After the tunnels are up, traffic is routed over the preferred tunnel. If the ASP associated with the preferred tunnel fails, the tunnel goes down and its traffic is automatically rerouted over the second tunnel. When the preferred tunnel recovers, its traffic is automatically switched back to the preferred tunnel.

Figure 10 illustrates the relationships between the various components of the example high availability configuration presented in this section. The example uses IKE proposals and IKE policies configured with the IKEv1 protocol; however, it applies equally to IKE proposals and IKE policies configured with the IKEv2 protocol.

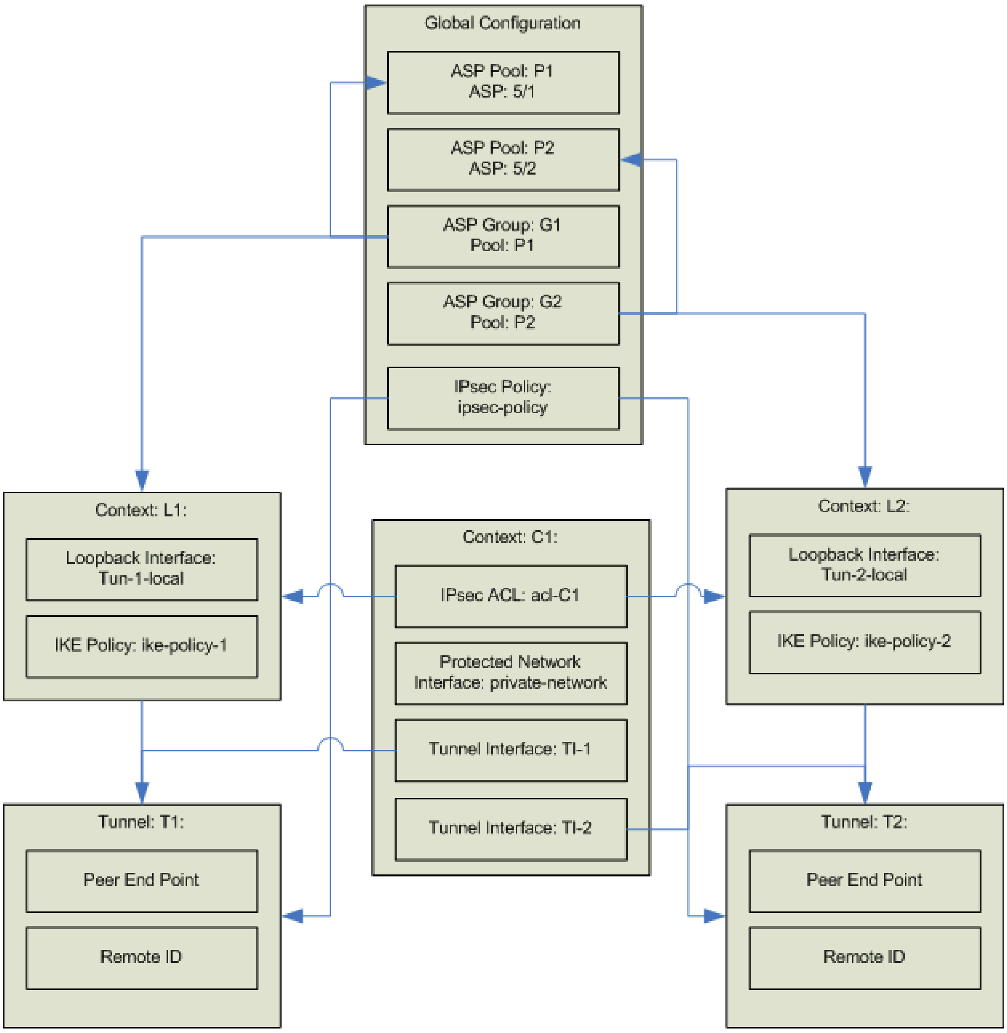

Figure 10 Relationships of the Components of a High Availability Configuration

To configure redundant IPsec tunnel connectivity between context C1 on the SmartEdge router and a remote security gateway:

1. Create the required ASP pools and groups, and two ASP groups.

   a Create two ASP pools, each with one ASP; for example, create ASP pool **P1** and enroll the first ASP on the ASE card in slot 5, and ASP pool **P2** and enroll the second ASP on the ASE card in slot 5:

```
[local]Redback(config)#asp pool P1 service security
[local]Redback(config-asp-pool-mode)#asp 5/1
[local]Redback(config-asp-pool-mode))#commit
[local]Redback(config)#asp pool P2 service security
[local]Redback(config-asp-pool-mode)#asp 5/2
[local]Redback(config-asp-pool-mode))#commit
[local]Redback(config-asp-pool-mode))#exit
```

b Create two ASP groups, point one ASP group to one ASP pool, and the other group to the other pool; for example, create ASP group `G1` and point it to ASP pool `P1`, and ASP group `G2` and point it to ASP pool `P2`:

```
[local]Redback(config)#asp group G1
[local]Redback(config-asp-group-mode)#pool P1
[local]Redback(config-asp-group-mode)#commit
[local]Redback(config-asp-group-mode)#exit
[local]Redback(config)#asp group G2
[local]Redback(config-asp-group-mode)#pool P2
[local]Redback(config-asp-group-mode)#commit
[local]Redback(config-asp-group-mode)#exit
```

2. Configure the required IKE proposals, IPsec proposals, and IPsec policies. For example:

```
[local]Redback(config)#ike proposal ike-proposal
[local]Redback(config-ike-proposal)#authentication algorithm
hmac-md5-96
[local]Redback(config-ike-proposal)#encryption algorithm
3des-cbc
[local]Redback(config-ike-proposal)#commit
[local]Redback(config-ike-proposal)#exit
[local]Redback(config)#ipsec proposal ipsec-proposal
[local]Redback(config-ipsec-proposal)#esp encryption
aes-128-cbc
[local]Redback(config-ipsec-proposal)#esp authentication
hmac-sha1-96
[local]Redback(config-ipsec-proposal)#commit
[local]Redback(config-ipsec-proposal)#exit
[local]Redback(config)#ipsec policy ipsec-policy
[local]Redback(config-ipsec-policy)#seq 10 proposal ipsec-
proposal
[local]Redback(config-ipsec-policy)#commit
[local]Redback(config-ipsec-policy)#exit
```

3. Create a context for the required interfaces, ACLs, and IP routes. For example, context `C1`:

```
[local]Redback(config)#context C1
[local]Redback(context)#commit
```

4. Create two contexts for the two IPsec tunnels. For example, contexts `L1` and `L2`:

```
[local]Redback(config)#context L1
[local]Redback(config)#context L2
[local]Redback(context)#commit
```

5. Configure contexts `L1` and `L2`. Create a loopback interface to identify the gateway, configure an IKE policy, enable the Security service, and

create a static route for signaling traffic. For example, in context `L1` create a loopback interface, `Tun-1-Local`, an IKE policy, `ike-policy1`, associate ASP group `G1` with this context to enable Security service, and specify an IP route for signaling traffic:

```
[local]Redback(config)#context L1

[local]Redback(config-ctx)#interface Tun-1-Local
  loopback
[local]Redback(config-if)#ip address 1.1.1.1/32
[local]Redback(config-if)#commit
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#ike policy ike-policy1
[local]Redback(config-ike-policy)#identity
  local 1.1.1.1
[local]Redback(config-ike-policy)#pre-shared-key
TEST_SHARED_KEY
[local]Redback(config-ike-policy)#seq 10 proposal
ike-proposal
[local]Redback(config-ike-policy)#commit
[local]Redback(config-ike-policy)#exit
[local]Redback(config-ctx)#asp-group G1
  service security
[local]Redback(config-ctx)#ip route 3.3.3.3/32 4.4.4.4
[local]Redback(config-ctx)#commit
[local]Redback(config-ctx)#exit
```

Repeat these configuration steps in context `L2`, create a loopback interface, `Tun-2-Local`, an IKE policy, `ike-policy2`, associate ASP group `G2` with this context to enable Security service, and specify an IP route for signaling traffic

6. Create the required tunnel and peer interfaces in context `C1`. For example, create two tunnel interfaces, `TI-1` and `TI-2`, and an interface to the protected network, `private-interface`:

```
[local]Redback(config)#context C1
[local]Redback(config-ctx)#interface TI-1
[local]Redback(config-if)#ip address 192.168.2.1/30
[local]Redback(config-if)#commit
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface TI-2
[local]Redback(config-if)#ip address 192.168.3.1/30
[local]Redback(config-if)#commit
[local]Redback(config-if)#exit
[local]Redback(config-ctx)#interface private-interface
[local]Redback(config-if)#ip address 172.168.3.0/24
[local]Redback(config-if)#commit
[local]Redback(config-if)#exit
```

7. Create an IPsec ACL in context `C1` to filter traffic between the two peers. For example, create the ACL **`acl-C1`:**

   ```
   [local]Redback(config)#context C1
   [local]Redback(config-ctx)#ipsec access-list acl-C1
   [local]Redback(config-ipsec-acl)#seq 10 172.168.2.0/24
   172.168.3.0/24
   ```

8. Create two IPsec tunnels, one in each security-enabled context. For example, create **`Tun-1`** in context **`L1`** and **`Tun-2`** in context **`L2`**:

   ```
   [local]Redback(config)#tunnel ipsec Tun-1
   [local]Redback(config-tunnel)#peer-end-point local 1.1.1.1
   remote 3.3.3.3 context L1
   [local]Redback(config-tunnel)#bind interface TI-1 C1
   [local]Redback(config-tunnel)#remote-id 3.3.3.3
   [local]Redback(config-tunnel)#ike-policy ike-policy1
   [local]Redback(config-tunnel)#seq 10 ipsec-policy
   ipsec-policy access-group acl-C1
   ```

   Repeat this step to create the second tunnel **`Tun-2`** in context **`L2`**, using appropriate values.

9. Create either static or dynamic IP routes between the remote protected network and each tunnel interface so that the traffic that flows over the tunnel is protected in context `C1`. Assign the IP routes over one tunnel a lower cost than the IP routes over the second tunnel. For example, create two static IP routes and assign a lower cost to IP routes on the tunnel interface **`TI-1`** used by **`Tun-1`** in context **`L1`** and a higher cost to IP routes learned on the tunnel interface **`TI-2`** used by **`Tun-2`** in context **`L2`**:

   ```
   [local]Redback(config)#context C1
   [local]Redback(config-ctx)#ip route 172.168.3.0/24 TI-1
   cost 5
   [local]Redback(config-ctx)#ip route 172.168.3.0/24 TI-2
   cost 10
   [local]Redback(config-ctx)#commit
   [local]Redback(config-ctx)#exit
   ```

While both tunnels are up, traffic is routed over the lower cost tunnel **`Tun-1`** in context **`L1`**. After the ASP associated with ASP group **`G1`** fails, **`Tun-1`** will go down and traffic will be automatically routed over **`Tun-2`**, until **`Tun-1`** recovers. Example 16 provides a configuration that matches the above procedure.

```
! Global asp pool config

asp pool P1 service security
asp 2/1
asp group G1
```

```
pool P1
asp-count 1

asp pool P2 service security
asp 2/2
asp group G2
pool P2
asp-count 1



context C1
interface TI-1
ip address 192.168.2.1/30

interface TI-2
ip address 192.168.3.1/30

interface private-interface
ip address 172.168.2.1/24

ipsec access-list acl-C1
seq 10 172.168.2.0/24 172.168.3.0/24

ip route 172.168.3.0/24 TI-1 cost 5
ip route 172.168.3.0/24 TI-2 cost 10




context L1
interface Tun-1-Local loopback
ip address 1.1.1.1/32

ike policy ike-policy1
identity local 1.1.1.1
pre-shared-key TEST_SHARED_KEY
seq 10 proposal ike-proposal

asp-group G1 service security

ip route 3.3.3.3/32 4.4.4.4

context L2
interface Tun-2-Local loopback
ip address 2.2.2.2/32

ike policy ike-policy2
identity local 2.2.2.2
pre-shared-key TEST_SHARED_KEY
seq 10 proposal ike-proposal
```

```
asp-group G2 service security

ip route 3.3.3.3/32 5.5.5.5



! Global Configuration

ike proposal ike-proposal
authentication algorithm hmac-md5-96
encryption algorithm 3des-cbc

ipsec proposal ipsec-proposal
esp encryption aes-128-cbc
esp authentication hmac-sha1-96

ipsec policy ipsec-policy
seq 10 proposal ipsec-proposal

tunnel ipsec Tun-1
peer-end-point local 1.1.1.1 remote 3.3.3.3 context L1
bind interface TI-1 C1
remote-id 3.3.3.3
ike-policy ike-policy1
seq 10 ipsec-policy ipsec-policy access-group acl-C1

tunnel ipsec Tun-2
peer-end-point local 2.2.2.2 remote 3.3.3.3 context L2
bind interface TI-2 C1
remote-id 3.3.3.3
ike-policy ike-policy2
seq 10 ipsec-policy ipsec-policy access-group acl-C1
```

Example 16   High Availability Configuration

14 IPsec VPN Restoration Following Service Disruption

IPsec VPN service is disrupted when a context associated with the Security service that is carrying IPsec VPN traffic or tunnels is missing or misconfigured.

Several scenarios can cause this type of service disruption:

- Following a catastrophic shutdown of the node, the node is restarted with an obsolete configuration.

  To resolve the situation, restore the latest backup of the configuration file for the SmartEdge router. If you regularly back up the configuration of the SmartEdge routers in your network to a secure location, the latest backup of the affected SmartEdge router may be more current than the latest version on the CompactFlash memory card. For more information, see *Managing Configuration Files* (Reference [8]).

- A context associated with Security services has been disassociated or deleted.

  If restoring the most recent backup is unsuccessful, identify the missing or misconfigured context and recreate the context with the correct settings.

- The physical link providing the underlying connectivity for an IPsec tunnel has been incorrectly modified or deleted.

  If restoring the most recent backup is unsuccessful, identify the missing or misconfigured link, then delete and recreate each IPsec tunnel that uses the link; this action takes down and brings up the endpoints used by each IPsec tunnel.

15 Log Messages Resulting from Rejected IKEv1 Packets

Table 5 lists the problems that can occur when processing IKEv1 policies and the log messages that identify them.

Table 5 Log Messages Resulting from Rejected IKEv1 Packets

Problem	Sample Message	Recommended Action
Mismatched preshared key	• `Payload Length(19136) is greater than the supported length(263)` • `Possibly Mismatching Keys detected in IKE phase-I 5th/6th messages` • `Sending phase-I notify of type UNEQUAL_PAYLOAD_LENGTHS`	Check the IKE policies defined at each endpoint to ensure that the preshared keys match.
Mismatched DH group	`Mismatching DH Group(2) detected, configured is (5) in IKE phase-I 1st message`	Check the IKE policies defined at each endpoint to ensure that the DH group values match.
Mismatched exchange type	• `Mismatching Xchange Type (MAIN_MODE) detected, configured is (AGGRESSIVE_MODE) in IKE phase-I 1st message` • `received unprotected message of notify type UNSUPPORTED_EXCHANGE_TYPE`	Check the IKE policies defined at each endpoint to ensure that the exchange types match.

Problem	**Sample Message**	**Recommended Action**
Mismatched authentication algorithm	`Mismatching Authentication algorithm (IKE_MD5) detected, configured is (IKE_SHA) in IKE phase-I 1st message.`	Check the IKE policies defined at each endpoint to ensure that the authentication algorithms match.
Mismatched encryption algorithm	`Mismatching Encryption algorithm (DES_CBC) detected, configured is (TRIPLE_DES_CBC) in IKE phase-I 1st message.`	Check the IKE policies defined at each endpoint to ensure that the encryption algorithms match.
Mismatched authentication method	`Mismatching Authentication method (PRE_SHAREDKEY) detected, configured is (RSA_SIGN) in phase-I message agent=IKE.`	Check the IKE policies defined at each endpoint to ensure that the authentication methods match.

Problem	Sample Message	Recommended Action
Mismatched ID data in the IKE policy when the authentication method uses digital certificates	• `ID type (ID_DER_ASN1_DN) and ID data /CN=ours not matching with the configured IKE policies` • `ID type (ID_IPV4_ADDR) and ID data 172.16.5.72 not matching with the configured IKE policies` • `ID type (ID_USER_FQDN) and ID data xz@x.com not matching with the configured IKE policies` • `ID type (ID_FQDN) and ID data dkoop.ours.com not matching with the configured IKE policies` • `NO CERT found that matches with at least 1 remoteid configured in IKE policy` • `Sending phase-I notify of type INVALID_ID_INFORMATION`	Check the IKE policies defined at each end to ensure that the ID data match when the authentication method uses digital certificates.

Problem	Sample Message	Recommended Action
Mismatched ID data for dial-in users in the IKE policy when the authentication method uses digital certificates	• `ID type (ID_DER_ASN1_DN) and ID data /CN=ours not matching with any of the configured IKE policies` • `ID type (ID_IPV4_ADDR) and ID data 172.16.5.72 not matching with any of the configured IKE policies` • `ID type (ID_USER_FQDN) and ID data xz@x.com not matching with any of the configured IKE policies` • `ID type (ID_FQDN) and ID data dkoop.ours.com not matching with any of the configured IKE policies` • `NO CERT found that matches with at least 1 remoteid configured in IKE policy` • `Sending phase-I notify of type INVALID_ID_INFORMATION`	Check the IKE policies defined at each end to ensure that the ID data for dial-in users match when the authentication method uses digital certificates.

Problem	Sample Message	Recommended Action
Mismatched ID data in the IKE policy when the authentication method uses a preshared key	• `Mismatching ID type (ID_IPV4_ADDR) and ID: 172.16.5.72 with the configured IKE policy agent=IKE` • `Mismatching ID type (ID_FQDN) and ID: dkoop.com with the configured IKE policy`	Check the IKE policies defined at each end to ensure that the ID data match when the authentication method uses a preshared key.
Mismatched ID payload with certificate identities	• `Mismatching ID type (ID_FQDN) and ID: dkoop.com with the configured IKE policy` • `ID type (ID_FQDN) and ID data:  dkoop.com mismatching with matching Cert ID type(ID_FQDN) and Cert ID: dkoop.ours$`	Check the IKE policies at both endpoints to ensure that the ID data match.
Mismatched ID data	`received NOTIFY PAYLOAD of notify type INVALID_ID_INFORMATION.`	Check the IKE policies at both endpoints to ensure that the ID data match.
First message in main mode contains an invalid next payload	• `Invalid Next Payload type(0x99) detected in IKE message` • `Sending phase-I notify of type INVALID_PAYLOAD_TYPE`	Check the IKE policies at both endpoints to ensure that the main mode attributes match.

Problem	Sample Message	Recommended Action
First message in main mode contains an invalid Digital Object Identifier (DOI) value	• `Invalid DOI(0x4001) detected in IKE phase-I 1st message` • `Sending phase-I notify of type DOI_NOT_SUPPORTED`	DOI used by IKE peer is invalid or not supported. Contact your support representative.
First message in main mode contains an invalid situation value	• `Situation(0x4001) is not supported` • `Sending phase-I notify of type INVALID_NOTIFY_TYPE`	Contact your support representative.
First message in main mode contains an invalid initiator cookie value	`Invalid Initiator Cookie detected in IKE message.`	Reset the VPN connection. If the problem persists, contact your support representative.
First message in main mode contains an invalid major version value in the Internet Security Association and Key Management Protocol (ISAKMP) header (either malformed or noncompliant)	• `Invalid Major Version(0x20) detected in IKE message` • `Sending phase-I notify of type INVALID_MAJOR_VERSION`	The version number is invalid. Contact your support representative.
First message in main mode contains an invalid minor version value in the ISAKMP header (either malformed or noncompliant)	• `Invalid Minor Version(0x2) detected in IKE message` • `Sending phase-I notify of type INVALID_MINOR_VERSION`	Contact your support representative.
First message in main mode contains an invalid exchange type	`Invalid Xchange Type(0x20) detected for non-existing IKE SA.`	Reset the VPN connection. If the problem persists, contact your support representative.

Problem	Sample Message	Recommended Action
First message in main mode contains invalid flags	• `Invalid Flags(0x f0) detected in IKE message` • `Sending phase-I notify of type INVALID_FLAGS`	Reset the VPN connection. If the problem persists, contact your support representative.
First message in main mode contains an invalid message ID value	• `Invalid Message ID(0x100100) detected in IKE phase-I message` • `Sending phase-I notify of type INVALID_MESSAGE_ID`	Contact your support representative.
First message in main mode contains an invalid protocol ID value	`Invalid Protocol ID(0xff) detected.`	Contact your support representative.
First message in main mode contains an invalid transform ID value	• `Invalid Xform ID(17) detected in IKE phase-I 1st message` • `Sending phase-I notify of type INVALID_TRANSFORM_ID`	Check the IKE policies at both endpoints to ensure that the main mode attributes match.
First message in main mode contains invalid attributes	• `Unsupported attribute(4099) detected in IKE phase-I 1st message` • `Sending phase-I notify of type ATTRIBUTES_NOT_SUPPORTED`	Check the IKE policies at both endpoints to ensure that the main mode attributes match.

Problem	Sample Message	Recommended Action
First message in main mode contains a mismatching proposal	• `Mismatching Encryption algorithm (TRIPLE_DES_CBC) detected, configured is (DES_CBC) in IKE phase-I 1st message` • `Sending phase-I notify of type NO_PROPOSAL_CHOSEN`	Check the IKE policies at both endpoints to ensure that the main mode attributes match.
First message in main mode contains an invalid reserved field value in the transform payload	• `Invalid Reserved field value(0x11) detected` • `Sending phase-I notify of type BAD_PROPOSAL_SYNTAX`	Check the IKE policies at both endpoints to ensure that the main mode attributes match.
First message in main mode contains a malformed payload	• `Invalid Reserved field (0x11) detected in IKE phase-1 1st message` • `Sending phase-I notify of type PAYLOAD_MALFORMED`	Check the IKE policies at both endpoints to ensure that the main mode attributes match.
First message in main mode contains an unsupported exchange type	`Unsupported XchgType detected.`	Check the IKE policies at both endpoints to ensure that the main mode attributes match.
First message in main mode contains a modified payload length in the SA payload	• `Payload Length(68) is greater than SA + Proposal + Xform + Attributes size(56)` • `Sending phase-I notify of type UNEQUAL_PAYLOAD_LENGTH`	Check the IKE policies at both endpoints to ensure that the main mode attributes match.

Problem	Sample Message	Recommended Action
Mismatched DH group in aggressive mode	`Mismatching DH Group(2) detected, configured is(5) in IKE phase-II 1st message.`	Check the IPsec policies defined at each endpoint to ensure that the DH group values match.
Mismatched authentication algorithm in aggressive mode	`Mismatching Authentication algorithm (MD5) detected, configured is (SHA1), in IKE phase-II 1st message .`	Check the IPsec policies defined at each endpoint to ensure that the authentication algorithms match.
Mismatched encryption algorithm in aggressive mode	`Mismatching Encryption algorithm (ESP_DES) detected, configured is (ESP_AES) in IKE phase-II message.`	Check the IPsec policies defined at each endpoint to ensure that the encryption algorithms match.
Selectors information sent in aggressive mode by the initiator	• `Initiator SPD selectors sent: IPADDR, 172.16.5.153, proto 1, port 0` • `Responder SPD selectors sent: IPADDR, 172.16.5.72, proto 1, port 0`	None. This is an informational log.
Selectors information received in aggressive mode by responder	• `Initiator SPD selectors rece ived: IPADDR, 172.16.5.72, proto 1, port 0` • `Responder SPD selectors rece ived: IPADDR, 172.16.5.153, proto 1, port 0`	None. This is an informational log.

Problem	Sample Message	Recommended Action
Mismatched selectors in aggressive mode	`No matching SPD policy for the selectors received in IKE phase-II message.`	Check the IPsec policies defined at each endpoint to ensure that the attributes match.
Mismatched security protocol in aggressive mode	• `Mismatching number of AH protocols recevied(1), configured number of AH protocols:  0 in IKE phase-II 1st message$` • `Mismatching number of ESP protocols recevied(0), configured number of ESP protocols:  1 in IKE phase-II 1st messa$`	Check the IPsec policies defined at each endpoint for ESP and AH values, as appropriate, to ensure that the attributes match
Lifetime value is in seconds for lifetime update	• `Notify payload of notify type RESPONDER_LIFETIME with protocol ESP or AH received` • `Received Notify Type RESPOND_NOTIFY with 300 seconds`	None. This is an informational log.

Problem	Sample Message	Recommended Action
Lifetime value is in kilobytes for lifetime update	• `Notify payload of notify type RESPONDER_LIFETIME with protocol ESP or AH received` • `Received Notify Type RESPOND_NOTIFY with 5 Kilo Bytes`	None. This is an informational log.
Reception of a delete payload request	• `Received DELETE PAYLOAD of protocol AH detected with SPI : 0x93a8d49d agent=IKE` • `Received DELETE PAYLOAD of protocol ESP detected with SPI : 0x9dd321d6`	None. This is an informational log.

16 Log Messages Resulting from Rejected IKEv2 Packets

Table 6 lists the problems that can occur when processing IKEv2 policies and the log messages that identify them.

Table 6 Log Messages Resulting from Rejected IKEv2 Packets

Type of Error	Sample Log Message
Authentication payload errors	• `AUTH PAYLOAD:Incomplete data received:1024` • `AUTH_PAYLOAD_LENGTH_ERROR :1024 bytes,received 512 bytes` • `AUTH PAYLOAD:Invalid AuthMethod Type:A`
Certificate payload errors	• `CERT PAYLOAD:Incomplete data received:2048 bytes` • `CERT_PAYLOAD_LENGTH_ERROR:200bytes,received 210 bytes` • `CERT_REQ__PAYLOAD_LENGTH_ERROR:200bytes,received 210 bytes` • `INVALID_CERT_ENCODING_TYPE _ERROR 22 in CERT payload`
CP Payload error	`CP_PAYLOAD:Incomplete data received:512 bytes`
Default payload errors	• `DEFAULT PAYLOAD:Incomplete data received:2048 bytes` • `DEFAULT PAYLOAD:Invalid Length received:2048 bytes`
Delete payload errors	• `DELETE_PAYLOAD_LENGTH_ERROR:1024 received 512 bytes` • `DELETE PAYLOAD:INVALID_PROTOCOL_ID:15` • `DELETE PAYLOAD:FOR IKE SA SPI SIZE IS NONZERO`

Type of Error	Sample Log Message
EAP payload errors	• EAP PAYLOAD:Incomplete data received:2048 bytes • EAP_PAYLOAD_LENGTH_ERROR:Invalid payload length:512 bytes
Encrypted payload errors	ENCRYPTED PAYLOAD:Encrypted payload MUST not be nested
ID payload errors	• ID PAYLOAD:Incomplete data received:512 bytes • ID_PAYLOAD_LENGTH_ERROR: usPayLoadLen 1024 bytes, ulMsgLen 512 bytes:
Key exchange payload errors	• KEY EXCHANGE PAYLOAD:Incomplete data received:2048 bytes • KEY_EXCHANGE_PAYLOAD_LENGTH_ERROR:200 bytes,received:210 bytes
Nonce payload errors	• NONCE PAYLOAD:INCOMPLETE_DATA_RECEIVED:2048 bytes • MAX_NONCE_PAYLOAD_LENGTH_ERROR:300,Maximum Nonce Payload size: 256+4
Notify payload errors	• NOTIFY PAYLOAD:INCOMPLETE_DATA_RECEIVED:512 bytes • NOTIFY_PAYLOAD_LENGTH_ERROR:1024,received 512 bytes: • NOTIFY PAYLOAD:INVALID_NOTIFY_PROTOCOL_ID:5 • NOTIFY PAYLOAD:FOR IKE SA SPI SIZE IS NONZERO
Proposal payload errors	PROPOSAL_PAYLOAD_LENGTH_ERROR : 1024,received bytes:512,
SA Payload errors	• SA PAYLOAD:Incomplete data received:1024 bytes • SA_PAYLOAD_LENGTH_ERROR:1024,received 512 bytes
Traffic selector payload error	ZERO_TRAFFIC_SELECTORS in TRAFFIC_SELECTOR_PAYLOAD

Type of Error	Sample Log Message
Unsupported payload errors	UNSUPPORTED PAYLOAD TYPE: 06, Critical Bit set
Vendor ID payload errors	• VENDOR ID PAYLOAD:Incomplete data received:2048 bytes • VENDOR ID PAYLOAD:Critical Bit Set • VENDOR_ID_PAYLOAD_LENGTH_ERROR:512 bytes
Xform payload errors	• INCOMPLETE_XFORM_PAYLOAD received :8 • XFORM_PAYLOAD_LENGTH_ERROR:1024 received bytes;512 • XFORM_ATTRIB_AF_Bit Set • XFORM_ATTRIB_LENGTH_ERROR:1250 bytes of Attribute Received:2400
IKEv2 attribute errors	• IKEV2_GENERIC_HEADER_LENGTH_ERROR:512 • IKEV2 HEADER:Incomplete data received:1024 bytes • IKEV2 HEADER:Zero Init SPI received • IKEV2 HEADER:INVALID_NEXT_PAYLOAD_TYPE • IKEV2 HEADER:Invalid Exchange Type • IKEV2_TOTAL_LENGTH_RCVD_ERROR:100 bytes IKE_HDR_TOTAL_LENGTH:110 • IKEV2_MAJOR_VERSION_ERROR:1
Incomplete attribute errors	Incomplete data received 512 bytes

Type of Error	Sample Log Message
Invalid attribute errors	• INVALID_ID_TYPE_ERROR:5 • INVALID_PROTOCOL_ID 4 in PROPOSAL_5, • INVALID_SPI_SIZE 0x3 in PROPOSAL_6 • INVALID_TRAFFIC_SELECTOR_TYPE:5 • INVALID_TRAFFIC_SELECTOR_LENGTH:7 • INVALID_CP_ATTRIB_TYPE:120 • INVALID_XFORM_ATTRIB_TYPE:12 Prop Num:5
Mismatched attribute errors	• ZERO_TRANSFORM received in PROPOSAL_6 • MISMATCHED_NUMBER_OF_SECURITY_PROTOCOLS:5 configured:6 in PROPOSAL_5 • MISMATCHED_ESN:10 configured:5 in PROPOSAL_5 • MISMATCHED_PFS_GROUP RECEIVED:2 configured:1 in PROPOSAL_5", • MISMATCHED_AH_AUTH_ALG:AUTH_NONE configured:HMAC-MD5 in PROPOSAL_5 • MISMATCHED_ESP_ENCRYPTION_ALG:AES-32 configured:3DES in PROPOSAL_5
Traffic selector attribute errors	Ikev2TSNtoH:INCOMPLETE_TRAFFIC_SELECTOR received:512 bytes

17 Log Messages Resulting from Rejected ESP Packets

Table 7 lists the problems that can occur when processing IPsec policies and the log messages that identify them.

Table 7 Log Messages Resulting from Rejected ESP Packets

Cause	Sample Messages	Recommended Action
Mismatched integrity check value in AH header	`Authentication failure. Possibly mis-matching keys.`	Check the IPsec policies defined at each endpoint to ensure that the attributes match
Mismatched authentication algorithm	`Authentication failure. Possibly mis-matching keys agent=IPSEC.`	Check the IPsec policies defined at each endpoint to ensure that the authentication algorithms match.
Mismatched integrity check value in ESP header	`Authentication failure. Possibly mis-matching keys.`	Check the IPsec policies defined at each endpoint to ensure that the attributes match.
Mismatched encapsulation mode	`Encapsulation mode differs. Received in Transport mode, where expected in Tunnel mode.`	Check the IPsec policies defined at each endpoint to ensure that the encapsulation modes (which must be Tunnel and not Transport) match.
ESP packet resent with the previous SPI	`No Inbound SA Found.`	None. This is an information log.
Late packet received with an offset greater the antireplay window size	`Late Packet falling out of window.`	None. This is an information log.
Repeated packet received	`Repeated/duplicate packet.`	None. This is an information log.

Glossary

ACL
Access Control List

AH
Authentication Header

CA
Certificate Authority

CoA
Change of Authorization

CPE
Customer Premises Equipment

DF
Don't Fragment

DH
Diffie-Hellman

DOI
Digital Object Identifier

DoS
Denial of Service

DPD
Dead Peer Detection

DSCP
Differentiated Services Code Point

ECMP
Equal-Cost Multipath

EMS
Element Management System

ESP
Encapsulating Security Payload

FQDNs
Fully Qualified Domain Names

IKE
Internet Key Exchange

IPsec
Internet Protocol Security

ISAKMP
Internet Security Association and Key Management Protocol

MD5
Mismatching Authentication algorithm

MTU
Maximum Transmission Unit

OSPF
Open Shortest Path First

P2MP
Point-to-Multipoint

PD
Priority Descriptor

PFS
Perfect Forward Secrecy

PKI
Public Key Infrastructure

PRF
Pseudo-Random Function

PSK
Preshared key

SA
Service Association

SAs
Security Associations

SPI
Security Parameter Index

VPN
Virtual Private Network

Reference List

[1] *Configuring Contexts and Interfaces*, 83/1543-CRA 119 1170/1-V1

[2] *SmartEdge and SM Node Configuration Guide*, 1543-CRA 119 1171/1

[3] *Configuring Basic IP Routing*, 16/1543-CRA 119 1170/1-V1

[4] *Configuring OSPF*, 22/1543-CRA 119 1170/1-V1

[5] *Configuring RIP*, 24/1543-CRA 119 1170/1-V1

[6] *Advanced Services Configuration and Operation Using the SmartEdge OS CLI*, 1/1543-CRA 119 1170/1-V1

[7] *IPsec VPN Command Reference*, 2/190 80-CRA 119 1170/1-V1

[8] *Managing Configuration Files*, 5/1543-CRA 119 1170/1-V1