JUNIPER NETWORKS

Junos® OS

IPSec Feature Guide

Release
12.1

Published: 2012-03-08

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

Junos® OS IPSec Feature Guide,
Release 12.1

Revision History
March 2012—R1 Junos OS 12.1

The information in this document is current as of the date on the title page.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at http://www.juniper.net/support/eula.html. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

List of Figures

List of Tables

PART 1

IPSec Features

CHAPTER 1

IPSec Concepts and Reference Material

This feature guide covers these topics:

Overview of IPSec

IP Security (IPSec) provides a secure way to authenticate senders and encrypt IP version 4 (IPv4) and version 6 (IPv6) traffic between network devices, such as routers and hosts. IPSec offers network administrators and their users the benefits of data confidentiality, data integrity, sender authentication, and anti-replay services. IPSec is increasingly becoming a critical component in today's contemporary IP networks.

IPSec is a framework for ensuring secure private communication over IP networks and is based on standards developed by the International Engineering Task Force (IETF). IPsec provides security services at the network layer of the Open Systems Interconnection (OSI) model by enabling a system to select required security protocols, determine the algorithms to use for the security services, and implement any cryptographic keys required to provide the requested services. You can use IPsec to protect one or more paths between

a pair of hosts, between a pair of security gateways (such as routers), or between a security gateway and a host.

The terminology and components of IPSec can be intimidating to first-time users. However, if you learn a few key concepts, you can quickly master and deploy IPSec in your network. The main concepts you need to understand are as follows:

IPSec-Enabled PICs

The first choice you need to make when implementing IPSec on a Junos OS-based router is the type of Physical Interface Card (PIC) you wish to use. There are three types of PICs available for M Series and T Series platforms:

- The ES PIC is a first-generation PIC that provides encryption services and software support for IPSec.
- The Adaptive Services (AS) PIC is a next-generation PIC that provides IPSec services and other services, such as Network Address Translation (NAT) and stateful firewall.
- The AS II Federal Information Processing Standards (FIPS) PIC is a special version of the AS PIC that communicates securely with the Routing Engine by using internal IPSec. You must configure IPSec on the AS II FIPS PIC when you enable FIPS mode on the router. For more information about implementing IPSec on an AS II FIPS PIC installed in a router configured in FIPS mode, see the *Secure Configuration Guide for Common Criteria and Junos-FIPS*.
- The MultiServices PICs supply hardware acceleration for an array of packet processing-intensive services in the M Series and T Series routers. These services include IPSec services and other services, such as stateful firewall, NAT, IPSec, anomaly detection, and tunnel services.

Authentication Algorithms

Authentication is the process of verifying the identity of the sender. Authentication algorithms use a shared key to verify the authenticity of the IPSec devices. The Junos OS uses the following authentication algorithms:

- Message Digest 5 (MD5) uses a one-way hash function to convert a message of arbitrary length to a fixed-length message digest of 128 bits. Because of the conversion process, it is mathematically infeasible to calculate the original message by computing it backwards from the resulting message digest. Likewise, a change to a single character in the message will cause it to generate a very different message digest number.

  To verify that the message has not been tampered with, the Junos OS compares the calculated message digest against a message digest that is decrypted with a shared key. The Junos OS uses the MD5 hashed message authentication code (HMAC) variant that provides an additional level of hashing. MD5 can be used with authentication header (AH), Encapsulating Security Payload (ESP), and Internet Key Exchange (IKE).

- Secure Hash Algorithm 1 (SHA-1) uses a stronger algorithm than MD5. SHA-1 takes a message of less than 264 bits in length and produces a 160-bit message digest. The large message digest ensures that the data has not been changed and that it originates from the correct source. The Junos OS uses the SHA-1 HMAC variant that provides an additional level of hashing. SHA-1 can be used with AH, ESP, and IKE.

- SHA-256, SHA-384, and SHA-512 (sometimes grouped under the name SHA-2) are variants of SHA-1 and use longer message digests. The Junos OS supports the SHA-256 version of SHA-2, which can process all versions of Advanced Encryption Standard (AES), Data Encryption Standard (DES), and Triple DES (3DES) encryption.

Encryption Algorithms

Encryption encodes data into a secure format so that it cannot be deciphered by unauthorized users. Like authentication algorithms, a shared key is used with encryption algorithms to verify the authenticity of the IPSec devices. The Junos OS uses the following encryption algorithms:

- Data Encryption Standard cipher-block chaining (DES-CBC) is a symmetric secret-key block algorithm. DES uses a key size of 64 bits, where 8 bits are used for error detection and the remaining 56 bits provide encryption. DES performs a series of simple logical operations on the shared key, including permutations and substitutions. CBC takes the first block of 64 bits of output from DES, combines this block with the second block, feeds this back into the DES algorithm, and repeats this process for all subsequent blocks.

- Triple DES-CBC (3DES-CBC) is an encryption algorithm that is similar to DES-CBC, but provides a much stronger encryption result because it uses three keys for 168-bit (3 x 56-bit) encryption. 3DES works by using the first key to encrypt the blocks, the second key to decrypt the blocks, and the third key to reencrypt the blocks.

- Advanced Encryption Standard (AES) is a next-generation encryption method based on the Rijndael algorithm developed by Belgian cryptographers Dr. Joan Daemen and

Dr. Vincent Rijmen. It uses a 128-bit block and three different key sizes (128, 192, and 256 bits). Depending on the key size, the algorithm performs a series of computations (10, 12, or 14 rounds) that include byte substitution, column mixing, row shifting, and key addition. The use of AES in conjunction with IPSec is defined in RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*.

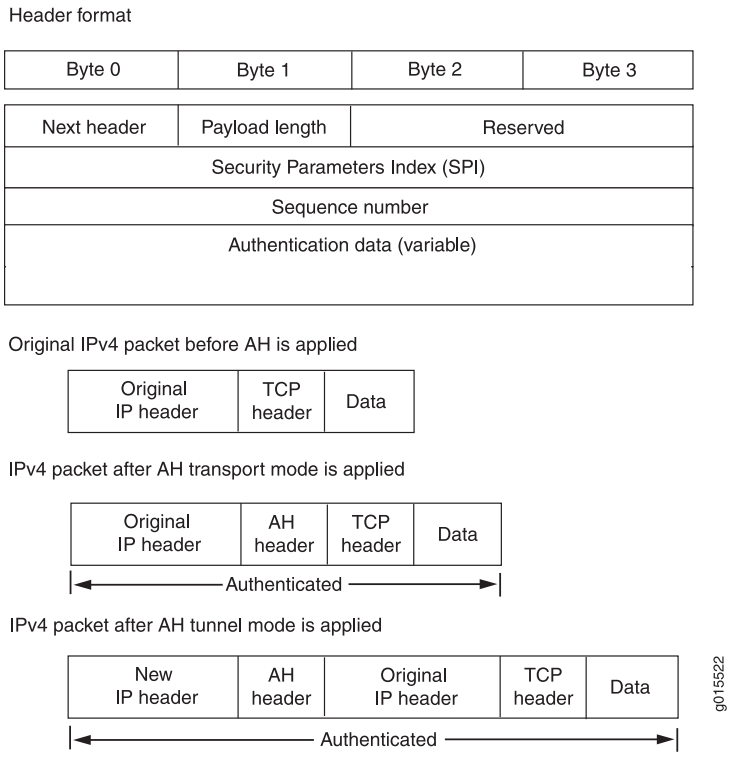
IPSec Protocols

IPSec protocols determine the type of authentication and encryption applied to packets that are secured by the router. The Junos OS supports the following IPSec protocols:

- AH—Defined in RFC 2402, AH provides connectionless integrity and data origin authentication for IPv4 and IPv6 packets. It also provides protection against replays. AH authenticates as much of the IP header as possible, as well as the upper-level protocol data. However, some IP header fields may change in transit. Because the value of these fields may not be predictable by the sender, they cannot be protected by AH. In an IP header, AH can be identified with a value of **51** in the **Protocol** field of an IPv4 packet and the **Next Header** field of an IPv6 packet. An example of the IPSec protection offered by AH is shown in Figure 1 on page 7.

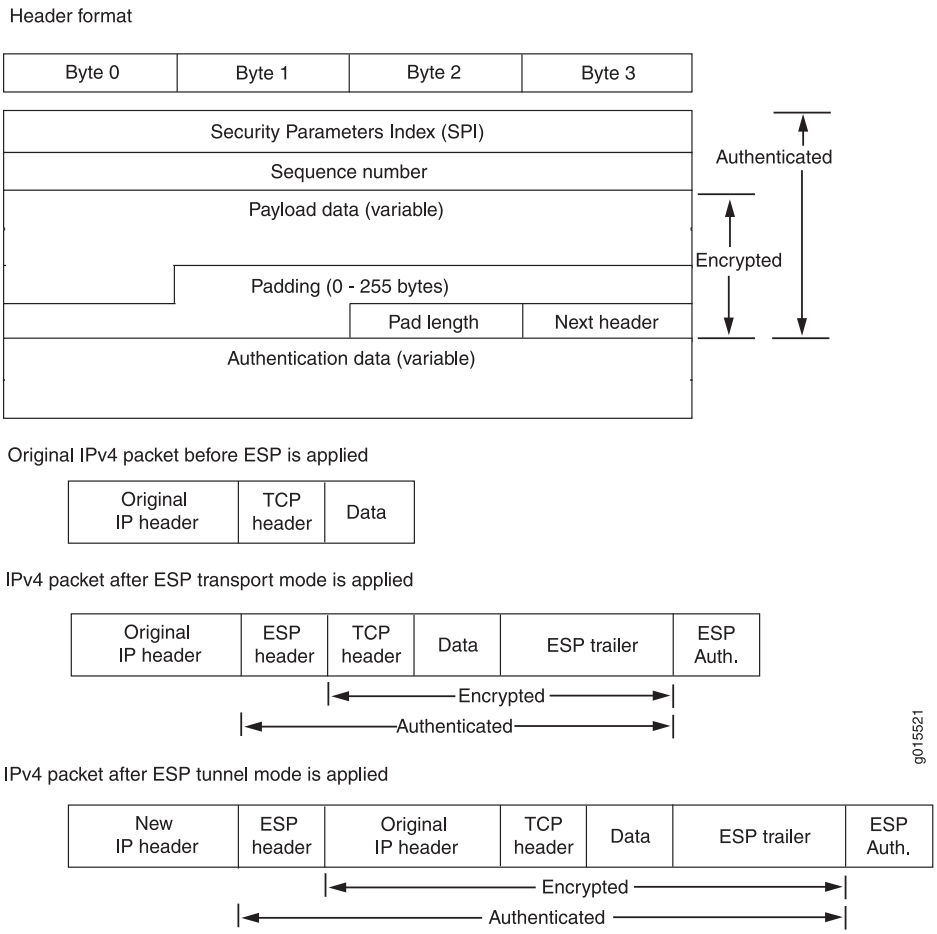
NOTE: **AH is not supported on the T Series, M120, and M320 routers.**

Figure 1: AH Protocol



- ESP—Defined in RFC 2406, ESP can provide encryption and limited traffic flow confidentiality, or connectionless integrity, data origin authentication, and an anti-replay service. In an IP header, ESP can be identified a value of **50** in the **Protocol** field of an IPv4 packet and the **Next Header** field of an IPv6 packet. An example of the IPSec protection offered by ESP is shown in Figure 2 on page 8.

Figure 2: ESP Protocol



- Bundle—When you compare AH with ESP, there are some benefits and shortcomings in both protocols. ESP provides a decent level of authentication and encryption, but does so only for part of the IP packet. Conversely, although AH does not provide encryption, it does provide authentication for the entire IP packet. Because of this, the Junos OS offers a third form of IPSec protocol called a protocol bundle. The bundle option offers a hybrid combination of AH authentication with ESP encryption.

Security Associations

Another IPSec consideration is the type of security association (SA) that you wish to implement. An SA is a set of IPSec specifications that are negotiated between devices that are establishing an IPSec relationship. These specifications include preferences for the type of authentication, encryption, and IPSec protocol that should be used when

establishing the IPSec connection. An SA can be either unidirectional or bidirectional, depending on the choices made by the network administrator. An SA is uniquely identified by a Security Parameter Index (SPI), an IPv4 or IPv6 destination address, and a security protocol (AH or ESP) identifier.

You can configure IPSec with a preset, preshared manual SA or use IKE to establish a dynamic SA. Manual SAs require you to specify all the IPSec requirements up front. Conversely, IKE dynamic SAs typically contain configuration defaults for the highest levels of authentication and encryption.

IPSec Modes

The last major consideration is the type of IPSec mode you wish to implement in your network. The Junos OS supports the following IPSec modes:

- Tunnel mode is supported for both AH and ESP in the Junos OS and is the usual choice for a router. In tunnel mode, the SA and associated protocols are applied to tunneled IPv4 or IPv6 packets. For a tunnel mode SA, an outer IP header specifies the IPsec processing destination, and an inner IP header specifies the ultimate destination for the packet. The security protocol header appears after the outer IP header, and before the inner IP header. In addition, there are slight differences for tunnel mode when you implement it with AH and ESP:
  - For AH, portions of the outer IP header are protected, as well as the entire tunneled IP packet.
  - For ESP, only the tunneled packet is protected, not the outer header.

  When one side of a security association is a security gateway (such as a router), the SA must use tunnel mode. However, when traffic (for example, SNMP commands or BGP sessions) is destined for a router, the system acts as a host. Transport mode is allowed in this case because the system does not act as a security gateway and does not send or receive transit traffic.
- Transport mode provides a security association between two hosts. In transport mode, the protocols provide protection primarily for upper layer protocols. For IPv4 and IPv6 packets, a transport mode security protocol header appears immediately after the IP header and any options, and before any higher layer protocols (for example, TCP or UDP). There are slight differences for transport mode when you implement it with AH and ESP:
  - For AH, selected portions of the IP header are protected, as well as selected portions of the extension headers and selected options within the IPv4 header.
  - For ESP, only the higher layer protocols are protected, not the IP header or any extension headers preceding the ESP header.

Digital Certificates

For small networks, the use of preshared keys in an IPSec configuration is often sufficient. However, as a network grows, it can become a challenge to add new preshared keys on

the local router and all new and existing IPSec peers. One solution for scaling an IPSec network is to use digital certificates.

A digital certificate implementation uses the public key infrastructure (PKI), which requires you to generate a key pair consisting of a public key and a private key. The keys are created with a random number generator and are used to encrypt and decrypt data. In networks that do not use digital certificates, an IPSec-enabled device encrypts data with the private key and IPSec peers decrypt the data with the public key.

With digital certificates, the key sharing process requires an additional level of complexity. First, you and your IPSec peers request a certificate authority (CA) to send you a CA certificate that contains the public key of the CA. Next, you request the CA to enroll a local digital certificate that contains your public key and some additional information. When the CA processes your request, it signs your local certificate with the private key of the CA. Then you install the CA certificate and the local certificate in your local router and load the CA certificate in the remote devices before you can establish IPSec tunnels with your peers.

When you request a peering relationship with an IPSec peer, the peer receives a copy of your local certificate. Because the peer already has the CA certificate loaded, it can use the CA's public key contained in the CA certificate to decrypt your local certificate that has been signed by the CA's private key. As a result, the peer now has a copy of your public key. The peer encrypts data with your public key before sending it to you. When your local router receives the data, it decrypts the data with your private key.

In the Junos OS, you must implement the following steps to be able to initially use digital certificates:

- Configure a CA profile to request CA and local digital certificates—The profile contains the name and URL of the CA or registration authority (RA), as well as some retry timer settings.
- Configure certificate revocation list support—A certificate revocation list (CRL) contains a list of certificates canceled before their expiration date. When a participating peer uses a CRL, the CA acquires the most recently issued CRL and checks the signature and validity of a peer's digital certificate. You can request and load CRLs manually, configure an LDAP server to handle CRL processing automatically, or disable CRL processing that is enabled by default.
- Request a digital certificate from the CA—The request can be made either online or manually. Online CA digital certificate requests use the Simple Certificate Enrollment Protocol (SCEP) format. If you request the CA certificate manually, you must also load the certificate manually.
- Generate a private/public key pair—The public key is included in the local digital certificate and the private key is used to decrypt data received from peers.
- Generate and enroll a local digital certificate—The local certificate can be processed online using SCEP or generated manually in the Public-Key Cryptography Standards

#10 (PKCS-10) format. If you create the local certificate request manually, you must also load the certificate manually.

- Apply the digital certificate to an IPSec configuration—To activate a local digital certificate, you configure the IKE proposal to use digital certificates instead of preshared keys, reference the local certificate in the IKE policy, and identify the CA in the service set.

Optionally, you can do the following:

- Configure the digital certificate to automatically reenroll—Starting in Junos OS Release 8.5, you can configure automatic reenrollment for digital certificates.
- Monitor digital certificate events and delete certificates and requests—You can issue operational mode commands to monitor IPSec tunnels established using digital certificates and delete certificates or requests.

For more details on managing digital certificates, configuring them in an IPSec service set, and monitoring and clearing them, see "Option: Using Digital Certificates" on page 26 and "Example: AS PIC IKE Dynamic SA with Digital Certificates Configuration" on page 90.

Service Sets

The Adaptive Services PIC supports two types of service sets when you configure IPSec tunnels. Because they are used for different purposes, it is important to know the differences between these service set types.

- Next-hop service set—Supports multicast and multicast-style dynamic routing protocols (such as OSPF) over IPSec. Next-hop service sets allow you to use *inside* and *outside* logical interfaces on the Adaptive Services PIC to connect with multiple routing instances. They also allow the use of Network Address Translation (NAT) and stateful firewall capabilities. However, next-hop service sets do not monitor Routing Engine traffic by default and require configuration of multiple service sets to support traffic from multiple interfaces.
- Interface service set—Applied to a physical interface and similar to a stateless firewall filter. They are easy to configure, can support traffic from multiple interfaces, and can monitor Routing Engine traffic by default. However, they cannot support dynamic routing protocols or multicast traffic over the IPSec tunnel.

In general, we recommend that you use next-hop service sets because they support routing protocols and multicast over the IPSec tunnel, they are easier to understand, and the routing table makes forwarding decisions without administrative intervention.

System Requirements

To implement IPSec, your system must meet these minimum requirements:

- Junos OS Release 8.5 or later for automatic reenrollment of digital certificates.
- Junos OS Release 8.3 or later for IPSec support on OSPF version 2
- Junos OS Release 8.2 or later for support on M120 routers

- Junos OS Release 8.1 or later for IPSec IKE support in routing instances, and certificate revocation list support on AS and MultiServices PICs installed on M Series and T Series routers
- Junos OS Release 7.6 or later for AES encryption and SHA-256 authentication support on AS PICs installed in M Series routers, and IPv6-based IPSec for AS PICs installed in M Series and T Series routers
- Junos OS Release 7.5 or later for digital certificate support on AS PICs installed in M Series and T Series routers, and support of the IPSec Monitoring Management Information Base (MIB)
- Junos OS Release 7.4 or later for dynamic endpoint tunneling support and configuring multiple routed tunnels in a single next-hop service set
- Junos OS Release 7.2 or later for transport mode IPSec on Routing Engines running OSPF version 3 and support for the AS II FIPS PIC
- Junos OS Release 7.1 or later for IPSec on the ES PIC for T Series and M320 routers
- Junos OS Release 6.4 or later for IPSec on the AS PIC for T Series and M320 routers
- Junos OS Release 6.2 or later for IPSec on the AS PIC for M Series routers
- Junos OS Release 5.7 or later for multicast over IPSec tunnels on M Series routers
- Junos OS Release 5.2 or later for IPSec on the ES PIC for M Series routers
- Two Juniper Networks M Series or T Series routers
- Two ES PICs or AS PICs for M Series and T Series routers

Terms and Acronyms

A

Adaptive Services PIC — A next-generation Physical Interface Card (PIC) that provides IPSec services and other services, such as Network Address Translation (NAT) and stateful firewall, on M Series and T Series platforms.

Advanced Encryption Standard (AES) — A next-generation encryption method that is based on the Rijndael algorithm and uses a 128-bit block, three different key sizes (128, 192, and 256 bits), and multiple rounds of processing to encrypt data.

authentication header (AH) — A component of the IPSec protocol used to verify that the contents of a packet have not changed (data integrity), and to validate the identity of the sender (data source authentication). For more information about AH, see RFC 2402.

C

certificate authority (CA) — A trusted third-party organization that generates, enrolls, validates, and revokes digital certificates. The CA guarantees the identity of a user and issues public and private keys for message encryption and decryption.

certificate revocation list (CRL)	A list of digital certificates that have been invalidated before their expiration date, including the reasons for their revocation and the names of the entities that have issued them. A CRL prevents usage of digital certificates and signatures that have been compromised.
cipher block chaining (CBC)	A cryptographic method that encrypts blocks of ciphertext by using the encryption result of one block to encrypt the next block. Upon decryption, the validity of each block of ciphertext depends on the validity of all the preceding ciphertext blocks. For more information on how to use CBC with DES and ESP to provide confidentiality, see RFC 2405.

D

Data Encryption Standard (DES)	An encryption algorithm that encrypts and decrypts packet data by processing the data with a single shared key. DES operates in increments of 64-bit blocks and provides 56-bit encryption.
digital certificate	Electronic file that uses private and public key technology to verify the identity of a certificate creator and distribute keys to peers.

E

Encapsulating Security Payload (ESP)	A component of the IPSec protocol used to encrypt data in an IPv4 or IPv6 packet, provide data integrity, and ensure data source authentication. For more information about ESP, see RFC 2406.
ES PIC	A PIC that provides first-generation encryption services and software support for IPSec on M Series and T Series platforms.

H

Hashed Message Authentication Code (HMAC)	A mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative cryptographic hash function, such as MD5 or SHA-1, in combination with a secret shared key. For more information on HMAC, see RFC 2104.

I

Internet Key Exchange (IKE)	Establishes shared security parameters for any hosts or routers using IPSec. IKE establishes the SAs for IPSec. For more information about IKE, see RFC 2407.

M

Message Digest 5 (MD5)	An authentication algorithm that takes a data message of arbitrary length and produces a 128-bit message digest. For more information, see RFC 1321.

P

Perfect Forward Secrecy (PFS)	Provides additional security by means of a Diffie-Hellman shared secret value. With PFS, if one key is compromised, previous and subsequent keys are secure because they are not derived from previous keys.
public key infrastructure (PKI)	A trust hierarchy that enables users of a public network to securely and privately exchange data through the use of public and private cryptographic key pairs that are obtained and shared with peers through a trusted authority.

R

registration authority (RA) — A trusted third-party organization that acts on behalf of a CA to guarantee the identity of a user.

Routing Engine — A PCI-based architectural portion of a Junos OS-based router that handles the routing protocol process, the interface process, some of the chassis components, system management, and user access.

S

Secure Hash Algorithm 1 (SHA-1) — An authentication algorithm that takes a data message of less than 264 bits in length and produces a 160-bit message digest. For more information on SHA-1, see RFC 3174.

Secure Hash Algorithm 2 (SHA-2) — A successor to the SHA-1 authentication algorithm that includes a group of SHA-1 variants (SHA-224, SHA-256, SHA-384, and SHA-512). SHA-2 algorithms use larger hash sizes and are designed to work with enhanced encryption algorithms such as AES.

security association (SA) — Specifications that must be agreed upon between two network devices before IKE or IPSec are allowed to function. SAs primarily specify protocol, authentication, and encryption options.

Security Association Database (SADB) — A database where all SAs are stored, monitored, and processed by IPSec.

Security Parameter Index (SPI) — An identifier that is used to uniquely identify an SA at a network host or router.

Security Policy Database (SPD) — A database that works with the SADB to ensure maximum packet security. For inbound packets, IPSec checks the SPD to verify if the incoming packet matches the security configured for a particular policy. For outbound packets, IPSec checks the SPD to see if the packet needs to be secured.

Simple Certificate Enrollment Protocol (SCEP) — A protocol that supports CA and registration authority (RA) public key distribution, certificate enrollment, certificate revocation, certificate queries, and certificate revocation list (CRL) queries.

T

Triple Data Encryption Standard (3DES) — An enhanced DES algorithm that provides 168-bit encryption by processing data three times with three different keys.

Configuring IPSec

To implement IPSec, complete the following configuration procedures:

- Considering General IPSec Issues on page 15
- Configuring Security Associations on page 19
- Using a Filter to Select Traffic to Be Secured on page 24
- Applying the Filter or Service Set to the Interface Receiving Traffic to Be Secured on page 25

Considering General IPSec Issues

Before you configure IPSec, it is helpful to understand some general guidelines.

- IPv4 and IPv6 traffic and tunnels—You can configure IPSec tunnels to carry traffic in the following ways: IPv4 traffic traveling over IPv4 IPSec tunnels, IPv6 traffic traveling over IPv4 IPSec tunnels, IPv4 traffic traveling over IPv6 IPSec tunnels, and IPv6 traffic traveling over IPv6 IPSec tunnels.
- Configuration syntax differences between the AS and MultiServices PICs and the ES PIC—There are slight differences in the configuration statements and operational mode commands that are used with the PICs that support IPSec. As a result, the syntax for the AS and MultiServices PICs cannot be used interchangeably with the syntax for the ES PIC. However, the syntax for one type of PIC can be converted to its equivalent syntax on the other PIC for interoperability. The differences are highlighted in Table 1 on page 15.
- Configuring keys for authentication and encryption—When preshared keys are required for authentication or encryption, you must use the guidelines shown in Table 2 on page 17 to implement the correct key size.
- Rejection of weak and semiweak keys—The DES and 3DES encryption algorithms will reject weak and semiweak keys. As a result, do not create and use keys that contain the patterns listed in Table 3 on page 17.

Table 1: Comparison of IPSec Configuration Statements and Operational Mode Commands for the AS and MultiServices PICs and ES PIC

AS and MultiServices PICs Statements and Commands	ES PIC Statements and Commands
Configuration Mode Statements	
[edit service-set *name*]	–

Table 1: Comparison of IPSec Configuration Statements and Operational Mode Commands for the AS and MultiServices PICs and ES PIC *(continued)*

AS and MultiServices PICs Statements and Commands	ES PIC Statements and Commands
[edit services ipsec-vpn ike] • policy {...} • proposal {...}	[edit security ike] • policy {...} • proposal {...}
[edit services ipsec-vpn ipsec] • policy {...} • proposal {...}	[edit security ipsec] • policy {...} • proposal {...}
[edit services ipsec-vpn rule *rule-name*] • remote-gateway *address*	[edit interface es- *fpc / pic /port*] • tunnel destination *address*
[edit services ipsec-vpn rule *rule-name* term *term-name*] • from *match-conditions* {...} then dynamic {...} • from *match-conditions* {...} then manual {...}	[edit security ipsec] • security-association *name* dynamic {...} • security-association *name* manual {...}
[edit services ipsec-vpn rule-set]	–
[edit services service-set ipsec-vpn] • local-gateway *address*	[edit interface es- *fpc /pic /port*] • tunnel source *address*
Operational Mode Commands	
clear security pki ca-certificate	–
clear security pki certificate-request	–
clear security pki local-certificate	–
clear services ipsec-vpn certificates	–
request security pki ca-certificate enroll	request security certificate (unsigned)
request security pki ca-certificate load	request system certificate add
request security pki generate-certificate-request	–
request security pki generate-key-pair	request security key-pair
request security pki local-certificate enroll	request security certificate (signed)

Table 1: Comparison of IPSec Configuration Statements and Operational Mode Commands for the AS and MultiServices PICs and ES PIC *(continued)*

AS and MultiServices PICs Statements and Commands	ES PIC Statements and Commands
request security pki local-certificate load	request system certificate add
show security pki ca-certificate	show system certificate
show security pki certificate-request	–
show security pki crl	–
show security pki local-certificate	show system certificate
show services ipsec-vpn certificates	show ipsec certificates
show services ipsec-vpn ike security-associations	show ike security-associations
show services ipsec-vpn ipsec security-associations	show ipsec security-associations

Table 2: Authentication and Encryption Key Lengths

	Number of Hexadecimal Characters	Number of ASCII Characters
Authentication		
HMAC-MD5-96	32	16
HMAC-SHA1-96	40	20
Encryption		
AES-128-CBC	16	32
AES-192-CBC	24	48
AES-256-CBC	32	64
DES-CBC	16	8
3DES-CBC	48	24

Table 3: Weak and Semiweak Keys

Weak Keys			
0101	0101	0101	0101

Table 3: Weak and Semiweak Keys *(continued)*

Weak Keys			
1F1F	1F1F	1F1F	1F1F
E0E0	E0E0	E0E0	E0E0
FEFE	FEFE	FEFE	FEFE
Semiweak Keys			
01FE	01FE	01FE	01FE
1FE0	1FE0	0EF1	0EF1
01E0	01E0	01F1	01F1
1FFE	1FFE	0EFE	0EFE
011F	011F	010E	010E
E0FE	E0FE	F1FE	F1FE
FE01	FE01	FE01	FE01
E01F	E01F	F10E	F10E
E001	E001	F101	F101
FEF1	FEF1	FE0E	FE0E
1F01	1F01	0E01	0E01
FEE0	FEE0	FEF1	FEF1

Keep in mind the following limitations of IPSec services on the AS PIC:

- The AS PIC does not transport packets containing IPv4 options across IPSec tunnels. If you try to send packets containing IP options across an IPSec tunnel, the packets are dropped. Also, if you issue a **ping** command with the **record-route** option across an IPSec tunnel, the **ping** command fails.
- The AS PIC does not transport packets containing the following IPv6 options across IPSec tunnels: hop-by-hop, destination (Type 1 and 2), and routing. If you try to send packets containing these IPv6 options across an IPSec tunnel, the packets are dropped.
- Destination class usage is not supported with IPSec services on the AS PIC.

Configuring Security Associations

The first IPSec configuration step is to select a type of security association for your IPSec connection. You must statically configure all specifications for manual SAs, but you can rely on some defaults when you configure an IKE dynamic SA. To configure a security association, see the following sections.

- Configuring Manual SAs on page 19
- Configuring IKE Dynamic SAs on page 20

Configuring Manual SAs

On the ES PIC, you configure a manual security association at the **[edit security ipsec security-association *name*]** hierarchy level. Include your choices for authentication, encryption, direction, mode, protocol, and SPI. Be sure that these choices are configured exactly the same way on the remote IPSec gateway.

```
[edit security]
ipsec {
  security-association sa-name {
    description description;
    manual {
      direction (inbound | outbound | bidirectional) {
        authentication {
          algorithm (hmac-md5-96 | hmac-sha1-96);
          key (ascii-text key | hexadecimal key);
        }
        auxiliary-spi auxiliary-spi;
        encryption {
          algorithm (des-cbc | 3des-cbc);
          key (ascii-text key  | hexadecimal key);
        }
        protocol (ah | esp | bundle);
        spi spi-value;
      }
    }
    mode (tunnel | transport);
  }
}
```

On the AS and MultiServices PICs, you configure a manual security association at the **[edit services ipsec-vpn rule *rule-name*]** hierarchy level. Include your choices for authentication, encryption, direction, protocol, and SPI. Be sure that these choices are configured exactly the same way on the remote IPSec gateway.

```
[edit services ipsec-vpn]
rule rule-name {
  match-direction (input | output);
  term term-name {
    from {
      destination-address address;
      source-address address;
    }
```

```
        then {
            backup-remote-gateway address;
            clear-dont-fragment-bit;
            manual {
                direction (inbound | outbound | bidirectional) {
                    authentication {
                        algorithm (hmac-md5-96 | hmac-sha1-96);
                        key (ascii-text key | hexadecimal key);
                    }
                    auxiliary-spi spi-value;
                    encryption {
                        algorithm algorithm; # This can be aes-128-cbc, aes-192-cbc,
                        # aes-256-cbc, des-cbc, or 3des-cbc.
                        key (ascii-text key | hexadecimal key);
                    }
                    protocol (ah | bundle | esp);
                    spi spi-value;
                }
            }
            no-anti-replay;
            remote-gateway address;
            syslog;
        }
    }
}
rule-set rule-set-name {
    [ rule rule-names ];
}
```

Configuring IKE Dynamic SAs

On the ES PIC, you configure an IKE dynamic SA at the **[edit security ike]** and **[edit security ipsec]** hierarchy levels. Include your choices for IKE policies and proposals, which include options for authentication algorithms, authentication methods, Diffie-Hellman groups, encryption, IKE modes, and preshared keys. The IKE policy must use the IP address of the remote end of the IPSec tunnel as the policy name. Also, include your choices for IPSec policies and proposals, which include options for authentication, encryption, protocols, Perfect Forward Secrecy (PFS), and IPSec modes. Be sure that these choices are configured exactly the same way on the remote IPSec gateway.

```
[edit security]
ike {
    proposal ike-proposal-name {
        authentication-algorithm (md5 | sha1);
        authentication-method (dsa-signatures | pre-shared-keys | rsa-signatures);
        description description;
        dh-group (group1 | group2);
        encryption-algorithm (3des-cbc | des-cbc);
        lifetime-seconds seconds;
    }
    policy ike-peer-address {
        description description;
        encoding (binary | pem);
        identity identity-name;
        local-certificate certificate-filename;
```

```
        local-key-pair private-public-key-file;
        mode (aggressive | main);
        pre-shared-key (ascii-text key | hexadecimal key);
        proposals [ proposal-names ];
    }
}
ipsec {
    proposal ipsec-proposal-name  {
        authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
        description description;
        encryption-algorithm (3des-cbc | des-cbc);
        lifetime-seconds seconds;
        protocol (ah | esp | bundle);
    }
    policy ipsec-policy-name {
        description description;
        perfect-forward-secrecy {
            keys (group1 | group2);
        }
        proposals [ proposal-names ];
    }
    security-association sa-name {
        description description;
        dynamic {
            ipsec-policy policy-name;
            replay-window-size (32 | 64);
        }
        mode (tunnel | transport);
    }
}
```

On the AS and MultiServices PICs, you configure an IKE dynamic security association at the **[edit services ipsec-vpn ike]**, **[edit services ipsec-vpn ipsec]**, and **[edit services ipsec-vpn rule *rule-name*]** hierarchy levels. Include your choices for IKE policies and proposals, which include options for authentication algorithms, authentication methods, Diffie-Hellman groups, encryption, IKE modes, and preshared keys. Also, include your choices for IPSec policies and proposals, which include options for authentication, encryption, protocols, PFS, and IPSec modes. Be sure that these choices are configured exactly the same way on the remote IPSec gateway.

If you choose not to explicitly configure IKE and IPSec policies and proposals on the AS and MultiServices PICs, your configuration can default to some preset values. These default values are shown in Table 4 on page 21.

Table 4: IKE and IPSec Proposal and Policy Default Values for the AS and MultiServices PICs

IKE Policy Statement	Default Value
mode	main
proposals	default
IKE Proposal Statement	Default Value

Table 4: IKE and IPSec Proposal and Policy Default Values for the AS and MultiServices PICs *(continued)*

IKE Policy Statement	Default Value
authentication-algorithm	sha1
authentication-method	pre-shared-keys
dh-group	group2
encryption-algorithm	3des-cbc
lifetime-seconds	3600 (seconds)
IPSec Policy Statement	**Default Value**
perfect-forward-secrecy keys	group2
proposals	default
IPSec Proposal Statement	**Default Value**
authentication-algorithm	hmac-sha1-96
encryption-algorithm	3des-cbc
lifetime-seconds	28800 (seconds)
protocol	esp

NOTE: If you use the default IKE and IPSec policy and proposal values preset within the AS and MultiServices PICs, you must explicitly configure an IKE policy and include a preshared key. This is because the **pre-shared-keys** authentication method is one of the preset values in the default IKE proposal.

If you decide to configure values manually, the following information shows the complete statement hierarchy and options for dynamic IKE SAs on the AS and MultiServices PICs:

```
[edit services ipsec-vpn]
ike {
  proposal proposal-name {
    authentication-algorithm (md5 | sha1 | sha256);
    authentication-method (pre-shared-keys | rsa-signatures);
    description description;
    dh-group (group1 | group2);
    encryption-algorithm algorithm; # This can be aes-128-cbc, aes-192-cbc,
    # aes-256-cbc, des-cbc, or 3des-cbc.
    lifetime-seconds seconds;
  }
```

```
    policy policy-name {
      description description;
      local-id {
        ipv4_addr [ values ];
        key_id [ values ];
      }
      local-certificate certificate-id-name;
      mode (aggressive | main);
      pre-shared-key (ascii-text key | hexadecimal key);
      proposals [ proposal-names ];
      remote-id {
        ipv4_addr [ values ];
        key_id [ values ];
      }
    }
  }
  ipsec {
    proposal proposal-name {
      authentication-algorithm (hmac-md5-96 | hmac-sha1-96);
      description description;
      encryption-algorithm algorithm; # This can be aes-128-cbc, aes-192-cbc,
      # aes-256-cbc, des-cbc, or 3des-cbc.
      lifetime-seconds seconds;
      protocol (ah | esp | bundle);
    }
    policy policy-name {
      description description;
      perfect-forward-secrecy {
        keys (group1 | group2);
      }
      proposals [ proposal-names ];
    }
  }
  rule rule-name {
    match-direction (input | output);
    term term-name {
      from {
        destination-address address;
        source-address address;
      }
      then {
        backup-remote-gateway address;
        clear-dont-fragment-bit;
        dynamic {
          ike-policy policy-name;
          ipsec-policy policy-name;
        }
        no-anti-replay;
        remote-gateway address;
        syslog;
      }
    }
  }
  rule-set rule-set-name {
    [ rule rule-names ];
  }
```

Using a Filter to Select Traffic to Be Secured

For the ES PIC, you need to configure a firewall filter to direct traffic into the IPSec tunnel. To apply a security association to traffic that matches a firewall filter, include the **ipsec-sa** *sa-name* statement at the **[edit firewall filter** *filter-name* **term** *term-name* **then]** hierarchy level.

```
[edit firewall filter filter-name]
term term-name {
  from {
    source-address {
      ip-address;
    }
    destination-address {
      ip-address;
    }
  }
  then {
    count counter-name;
    ipsec-sa sa-name;
  }
}
term other {
  then accept;
}
```

For the AS and MultiServices PICs, you do not need to configure a separate firewall filter. A filter is already built into the IPSec VPN **rule** statement at the **[edit services ipsec-vpn]** hierarchy level. To apply a security association to traffic that matches the IPSec VPN rule, include the **dynamic** or **manual** statement at the **[edit services rule** *rule-name* **term** *term-name* **then]** hierarchy level. To specify whether the rule should match input or output traffic, include the **match-direction** statement at the **[edit services rule** *rule-name***]** hierarchy level.

After defining the rules for your IPSec VPNs, you must apply the rules to a service set. To do this, include the **ipsec-vpn-rules** *rule-name* statement at the **[edit services service-set** *service-set-name***]** hierarchy level. Include an IPv4 or IPv6 IPSec gateway with the **local-gateway** *local-ip-address* statement at the **[edit services service-set** *service-set-name***]** hierarchy level.

Also, you must select either a single interface or a pair of interfaces that participate in IPSec. To select a single interface, include the interface-service ***interface-name*** statement at the **[edit services service-set** *service-set-name***]** hierarchy level. To select a pair of interfaces and a next hop, include the **next-hop-service** statement at the **[edit services service-set** *service-set-name***]** hierarchy level and specify an inside interface and an outside interface. Only next-hop service sets support IPSec within Layer 3 VPNs and use of routing protocols over the IPSec tunnel.

```
[edit services]
service-set service-set-name {
  interface-service {
    service-interface interface-name;
```

```
    }
    next-hop-service {
      inside-service-interface interface-name;
      outside-service-interface interface-name;
    }
    ipsec-vpn-options {
      local-gateway local-ip-address <routing-instance instance-name>;
      trusted-ca ca-profile-name;
    }
    ipsec-vpn-rules rule-name;
  }
  ipsec-vpn {
    rule rule-name {
      term term-name {
        from {
          source-address {
            ip-address;
          }
          destination-address {
            ip-address;
          }
        }
        then {
          remote-gateway remote-ip-address;
          (dynamic | manual);
        }
      }
      match-direction output;
    }
  }
```

Applying the Filter or Service Set to the Interface Receiving Traffic to Be Secured

For the ES PIC, apply your firewall filter on the input interface receiving the traffic that you wish to send to the IPSec tunnel. To do this, include the **filter** statement at the **[edit interfaces** *interface-name* **unit** *unit-number* **family inet]** hierarchy level.

```
[edit interfaces interface-name unit unit-number family inet]
filter {
  input filter-name;
}
```

For the AS and MultiServices PICs, apply your IPSec-based interface service set to the input interface receiving the traffic that you wish to send to the IPSec tunnel. To do this, include the **service-set** *service-set-name* statement at the **[edit interfaces** *interface-name* **unit** *unit-number* **family inet service (input | output)]** hierarchy level.

```
[edit interfaces interface-name unit unit-number family inet]
service {
  input {
    service-set service-set-name;
  }
  output {
    service-set service-set-name;
  }
```

```
}
```

To configure a next-hop-based service set on the AS and MultiServices PICs, include the **service-domain** statement at the **[edit interfaces *interface-name* unit *unit-number*]** hierarchy level and specify one logical interface on the AS PIC as an inside interface and a second logical interface on the AS PIC as an outside interface.

```
[edit interfaces sp-fpc/pic/port]
unit 0 {
   family inet {
      address ip-address;
   }
}
unit 1 {
   family inet;
   service-domain inside;
}
unit 2 {
   family inet;
   service-domain outside;
}
```

Option: Using Digital Certificates

A popular way for network administrators to scale an IPSec network is to use digital certificates instead of preshared keys. To enable digital certificates in your network, you need to use a combination of operational mode commands and configuration statements. The following steps enable you to implement digital certificates on AS and MultiServices PICs installed in M Series and T Series routers:

Configuring a CA Profile

The CA profile contains the name and URL of the CA or RA, as well as some retry timer settings. CA certificates issued by Entrust, VeriSign, and Microsoft are all compatible with M Series, and T Series routers. To configure the domain name of the CA or RA, include the **ca-identity** statement at the **[edit security pki ca-profile *ca-profile-name*]** hierarchy level. To configure the URL of the CA, include the **url** statement at the **[edit security pki**

ca-profile *ca-profile-name* **enrollment]** hierarchy level. To configure the number of enrollment attempts the router should perform, include the **retry** statement at the **[edit security pki ca-profile** *ca-profile-name* **enrollment]** hierarchy level. To configure the amount of time the router should wait between enrollment attempts, include the **retry-interval** statement at the **[edit security pki ca-profile** *ca-profile-name* **enrollment]** hierarchy level.

```
[edit security pki]
ca-profile ca-profile-name {
    ca-identity ca-identity;
    enrollment {
        url url-name;
        retry number-of-enrollment-attempts; # The range is 0 though 100 attempts.
        retry-interval seconds; # The range is 0 though 3600 seconds.
    }
}
```

Configuring a Certificate Revocation List

A certificate revocation list (CRL) contains a list of digital certificates that have been canceled before their expiration date. When a participating peer uses a digital certificate, it checks the certificate signature and validity. It also acquires the most recently issued CRL and checks that the certificate serial number is not on that CRL. By default, CRL verification is enabled on any CA profile running on Junos OS Release 8.1 or later. To disable CRL verification, include the **disable** statement at the [**edit security pki ca-profile** *ca-profile-name* **revocation-check**] hierarchy level.

To specify the URL for the Lightweight Directory Access Protocol (LDAP) server where your CA stores its current CRL, include the **url** statement at the [**edit security pki ca-profile** *ca-profile-name* **revocation-check crl**] hierarchy level. If the LDAP server requires a password to access the CRL, include the **password** statement at the [**edit security pki ca-profile** *ca-profile-name* **revocation-check crl url**] hierarchy level.

NOTE: You do not need to specify a URL for the LDAP server if the certificate includes a certificate distribution point (CDP). The CDP is a field within the certificate that contains information about how to retrieve the CRL for the certificate. The router uses this information to download the CRL automatically. Any LDAP URL you configure takes precedence over the CDP included in the certificate.

If you manually downloaded the CRL, you must manually install it on the router. To manually install the CRL, issue the **request security pki crl load ca-profile** *ca-profile-name* **filename** *path/filename* command.

To configure the time interval between CRL updates, include the **refresh-interval** statement at the [**edit security ca-profile** *ca-profile-name* **revocation-check crl**] hierarchy level.

To override the default behavior and permit IPSec peer authentication to continue when the CRL fails to download, include the **disable on-download-failure** statement at the [**edit security ca-profile** *ca-profile-name* **revocation-check crl**] hierarchy level.

```
[edit security pki ca-profile ca-profile-name]
revocation-check {
  disable;
  crl {
    disable on-download-failure;
    refresh-interval number-of-hours { # The range is 0 through 8784 hours.
      url {
        url-name;
        password;
      }
    }
  }
}
```

Requesting a CA Digital Certificate

You can request a CA digital certificate either online or manually. To request a digital certificate from a CA or RA online by using SCEP, issue the **request security pki ca-certificate enroll ca-profile** *ca-profile-name* command.

If you obtained the CA digital certificate manually through e-mail or other out-of-band mechanism, you must load it manually. To manually install a certificate in your router, issue the **request security pki local-certificate load** command.

Generating a Private/Public Key Pair

A key pair is a critical element of a digital certificate implementation. The public key is included in the local digital certificate and the private key is used to decrypt data received from peers. To generate a private/public key pair, issue the **request security pki generate-key-pair certificate-id** *certificate-id-name* command.

Generating and Enrolling a Local Digital Certificate

You can generate and enroll a local digital certificate either online or manually. To generate and enroll a local certificate online by using SCEP, issue the **request security pki local-certificate enroll** command. To generate a local certificate request manually in the PKCS-10 format, issue the **request security pki generate-certificate-request** command.

If you create the local certificate request manually, you must also load the certificate manually. To manually install a certificate in your router, issue the **request security pki local-certificate load** command.

Applying the Local Digital Certificate to an IPSec Configuration

To activate a local digital certificate, you configure the IKE proposal to use digital certificates instead of preshared keys, reference the local certificate in the IKE policy, and identify the CA or RA in the service set. To enable the IKE proposal for digital certificates, include the **rsa-signatures** statement at the **[edit services ipsec-vpn ike proposal** *proposal-name* **authentication-method]** hierarchy level. To reference the local certificate in the IKE policy, include the **local-certificate** statement at the **[edit services ipsec-vpn ike policy** *policy-name***]** hierarchy level. To identify the CA or RA in the service

set, include the **trusted-ca** statement at the **[edit services service-set *service-set-name* ipsec-vpn-options]** hierarchy level.

```
[edit services]
service-set service-set-name {
    .....
    ipsec-vpn-options {
        trusted-ca ca-profile-name;
    }
}
ipsec-vpn {
    ike {
        proposal proposal-name  {
            .....
            authentication-method [pre-shared-keys | rsa-signatures];
        }
        policy policy-name {
            ....
            local-certificate certificate-id-name;
        }
    }
}
```

Configuring Automatic Reenrollment of Digital Certificates

You can configure automatic reenrollment for digital certificates. This feature is by default not enabled. To configure automatic reenrollment for digital certificates, include the **auto-re-enrollment** statement at the [**edit security pki**] hierarchy level:

```
[edit]
security {
    pki {
        auto-re-enrollment {
            certificate-id certificate-name {
                ca-profile ca-profile-name;
                challenge-password password;
                re-enroll-trigger-time-percentage percentage; # Percentage of validity-period
                    # (specified in certificate) when automatic
                    # reenrollment should be initiated.
                re-generate-keypair;
                validity-period number-of-days;
            }
        }
    }
}
```

Monitoring Digital Certificates

Purpose You can issue various forms of the **show security pki** command to view digital certificates and certificate requests and certificate revocation lists:

Action

- To display the CA digital certificate, issue the **show security pki ca-certificate ca-profile** ***ca-profile-name*** command.
- To display the local digital certificate and the public key used to enroll the certificate, issue the **show security pki local-certificate certificate-id** ***certificate-id-name*** command.
- To display the local certificate request in PKCS-10 format, issue the **show security pki certificate-request certificate-id** ***certificate-id-name*** command.
- You can also view which digital certificates are used in IKE negotiations to establish IPSec tunnels by issuing the **show services ipsec-vpn certificates** command.
- To display the certificate revocation list, issue the **show security pki crl ca-profile** ***ca-profile-name*** command.
- To determine if a certificate is enabled for automatic-reenrollment, issue the **show security pki** command.

Clearing Digital Certificates

Purpose Variations of the **clear security pki** command enable you to delete certificates or requests and certificate revocation lists:

Action

- To delete the CA digital certificate, issue the **clear security pki ca-certificate ca-profile** ***ca-profile-name*** command.
- To delete the local digital certificate and the associated private/public key pair, issue the **clear security pki local-certificate certificate-id** ***certificate-id-name*** command.
- To delete the local certificate request, issue the **clear security pki certificate-request certificate-id** ***certificate-id-name*** command.
- To clear the digital certificates that were used in IKE negotiations to establish IPSec tunnels, issue the **clear services ipsec-vpn certificates** command.
- To delete the certificate revocation list, issue the **clear security pki crl ca-profile** ***ca-profile-name*** command.

Related Documentation

- Example: AS PIC IKE Dynamic SA with Digital Certificates Configuration on page 90
- IP Security Operational Mode Commands
- System Basics: Security Services Configuration Guide
- IPsec Properties

Option: Using Filter-Based Forwarding to Select Traffic to Be Secured

Instead of using a firewall filter, you can also forward traffic into an IPSec security association by using a filter-based forwarding instance. First, configure the filter-based forwarding instance. Then, configure a routing table group to advertise the routes from the filter-based forwarding instance. Next, create a firewall filter for the ES PIC and

reference the filter-based forwarding instance. Lastly, apply the filter and IPSec security association to the ES PIC.

```
[edit]
routing-instances {
   forwarding {
      instance-type forwarding;
      routing-options {
         static {
            route 10.10.10.0/24 next-hop 192.168.0.5;
         }
      }
   }
}
routing-options {
   rib-groups {
      group-name  {
         import-rib [ inet.0 forwarding.inet.0 ];
      }
   }
}
firewall {
   family inet {
      filter filter-name {
         term term-name {
            then routing-instance instance-name;
         }
      }
   }
}
[edit]
interfaces {
   es-0/0/0 {
      unit 0 {
         tunnel {
            source source-ip-address;
            destination destination-ip-address;
         }
         family inet {
            ipsec-sa sa-name;
            filter {
               input filter-name;
            }
            address ip-address;
         }
      }
   }
}
```

Option: Using IPSec with a Layer 3 VPN

Some key concepts to keep in mind when configuring IPSec within a VPN include the following:

- Add the outside services interface for a next-hop style service set into the routing instance by including the **interface sp-*fpc/pic/port*** statement at the **[edit routing-instances *instance-name*]** hierarchy level.
- For interface style service sets, add the interface on which you apply the service set and the services interface by including both interfaces at the **[edit routing-instances *instance-name*]** hierarchy level.
- To define a routing instance for the local gateway within the service set, include the **routing-instance *instance-name*** option at the **[edit services service-set *service-set-name* ipsec-vpn-options local-gateway *address*]** hierarchy level.

The following configuration for an AS PIC on a provider edge (PE) router demonstrates the use of next-hop service sets with an IKE dynamic SA in a VPN routing and forwarding (VRF) routing instance.

```
[edit]
interfaces {
   so-0/0/0 {
      description "Interface connected to the customer edge (CE) router";
      unit 0 {
         family inet {
            address 10.6.6.6/32;
         }
      }
   }
   so-2/2/0 {
      description "Source IPSec tunnel interface to the network core";
      unit 0 {
         family inet {
            address 10.10.1.1/30;
         }
      }
   }
   sp-3/1/0 {
      description "AS PIC interface";
      unit 0 {
         family inet {
            address 10.7.7.7/32;
         }
      }
      unit 1 {
         family inet;
         service-domain inside;
      }
      unit 2 {
         family inet;
         service-domain outside;
      }
```

```
    }
}
policy-options {
    policy-statement vpn-export-policy {
        then {
            community add community-name;
            accept;
        }
    }
    policy-statement vpn-import-policy {
        term term-name {
            from community community-name;
            then accept;
        }
    }
    community community-name members target:100:20;
}
routing-instances {
    vrf {
        instance-type vrf;
        interface sp-3/1/0.1; # Inside sp interface.
        interface so-0/0/0.0; # Interface that connects to the CE router.
        route-distinguisher route-distinguisher;
        vrf-import vpn-import-policy;
        vrf-export vpn-export-policy;
        routing-options {
            static {
                route ip-address/prefix next-hop so-0/0/0.0; # Routes for the CE router.
                route ip-address/prefix next-hop sp-3/1/0.1; # Routes for IPSec.
            }
        }
    }
}
services {
    service-set service-set-name {
        next-hop-service {
            inside-service-interface sp-3/1/0.1;
            outside-service-interface sp-3/1/0.2;
        }
        ipsec-vpn-options {
            local-gateway 10.10.1.1;
        }
        ipsec-vpn-rules rule-name;
    }
    ipsec-vpn {
        rule rule-name {
            term term-name {
                from {
                    source-address {
                        source-ip-address;
                    }
                }
                then {
                    remote-gateway 10.10.1.2;
                    dynamic {
                        ike-policy ike-policy-name;
```

```
            }
          }
        }
        match-direction direction;
      }
      ike {
        policy ike-policy-name {
          pre-shared-key ascii-text preshared-key;
        }
      }
    }
  }
```

For more information on VRF routing instances, see the *Junos VPNs Configuration Guide*. For more information on next-hop service sets, see the *Junos Services Interfaces Configuration Guide*.

Option: Securing BGP Sessions with Transport Mode

For the ES PIC, you can use IPSec to secure BGP sessions between Routing Engines in M Series and T Series platforms. To configure, create a transport mode security association and apply the SA to the BGP configuration by including the **ipsec-sa** statement at the **[edit protocols bgp group *group-name*]** hierarchy level.

```
[edit]
protocols {
  bgp {
    group group-name {
      local-address ip-address;
      export export-policy;
      peer-as as-number;
      ipsec-sa sa-name;
      neighbor peer-ip-address;
    }
  }
}
```

Option: Securing OSPFv3 Networks with Transport Mode

OSPF version 3 (OSPFv3), unlike OSPF version 2, does not have a built-in authentication method and relies on IPSec to provide this functionality. Using the ES PIC syntax, you can use IPSec to secure OSPFv3 between Routing Engines in M Series and T Series platforms. You can secure specific OSPFv3 interfaces and protect OSPFv3 virtual links. To configure, create a transport mode security association and apply the SA to the OSPFv3 configuration by including the **ipsec-sa** statement at the **[edit protocols ospf3 area *area-number* interface *interface-name*]** or **[edit protocols ospf3 area *area-number* virtual-link neighbor-id *neighbor-ip-address* transit-area *area-number*]** hierarchy level.

```
[edit]
protocols {
  ospf3 {
    area area-number {
      interface interface-name {
        ipsec-sa sa-name;
```

```
        }
        virtual-link neighbor-id neighbor-ip-address transit-area area-number {
          ipsec-sa sa-name;
        }
      }
    }
  }
```

Option: Securing OSPFv2 Networks with Transport Mode

By default, you can configure MD5 or simple text password-based authentication over OSPFv2 links. In addition to these basic authentications, the Junos OS supports OSPFv2 with a security authentication header (AH), Encapsulating Security Payload (ESP), or an IPSec protocol bundle that supports both AH and ESP. You can configure IPSec over OSPFv2 using transport mode security associations on physical, sham, or virtual links.

Because the Junos OS supports only bidirectional security associations over OSPFv2, OSPFv2 peers must be configured with the same IPSec security association. Configuring OSPFv2 peers with different security associations or with dynamic IKE will prevent adjacencies from being established. In addition, you must configure identical security associations for sham links with the same remote endpoint address, for virtual links with the same remote endpoint address, for all neighbors on OSPF nonbroadcast multiaccess (NBMA) or point-to-multipoint links, and for every subnet that is part of a broadcast link.

To create a manual bidirectional security association, include the **security-association** ***security-association-name*** statement at the [**edit security ipsec**] hierarchy level:

```
[edit]
security {
  ipsec {
    security-association security-association name {
      mode transport;
      manual {
        direction bidirectional {
          protocol (ah | esp | bundle);
          spi spi--value;
          authentication {
            algorithm (hmac-md5-96 | hmac-sha1-96);
            key (ascii-text key | hexadecimal key);
          }
        }
      }
    }
  }
}
```

To configure IPSec on an OSPFv2 interface, create a transport mode security association and include the **ipsec-sa name** statement at the [**edit protocols ospf area** ***area-id***] hierarchy level:

```
[edit]
protocols {
```

```
ospf {
  area area-id {
    interface interface-name {
      ipsec-sa sa-name;
    }
    virtual-link neighbor-id a.b.c.d transit-area x.x.x.x {
      ipsec-sa sa-name;
    }
    sham-link-remote {
      ipsec-sa sa-name;
    }
  }
}
}
```

To verify your configuration, enter the **show ospf interface detail** command. This command gives detailed information about the **ospfv2** interface and displays the interface's security association at the bottom of the output. In the example below, the security association configured on this router is **sa1**.

```
user@router> show ospf interface detail
Interface              State     Area            DR ID           BDR ID Nbrs
fe-0/0/1.0             BDR      0.0.0.0         192.168.37.12   10.255.245.215 1
Type LAN, address 192.168.37.11, Mask 255.255.255.248, MTU 4460, Cost 40
DR addr 192.168.37.12, BDR addr 192.168.37.11, Adj count 1, Priority 128
Hello 10, Dead 40, ReXmit 5, Not Stub
t1-0/2/1.0             PtToPt   0.0.0.0         0.0.0.0         0.0.0.0 0
Type P2P, Address 0.0.0.0, Mask 0.0.0.0, MTU 1500, Cost 2604
Adj count 0
Hello 10, Dead 40, ReXmit 5, Not Stub
Auth type: MD5, Active key ID 3, Start time 2002 Nov 19 10:00:00 PST
IPsec SA Name: sa1
```

Option: Monitoring IPSec by Using SNMP

In Junos OS Release 7.5 and later, the IPSec Monitoring MIB provides a way to monitor IPSec information on AS PICs installed in M Series and T Series routers by using the Simple Network Management Protocol (SNMP). The MIB provides an IKE tunnel table to monitor IKE security associations and view related statistics, an IPSec tunnel table to view IPSec tunnel statistics, and an IPSec security associations table to view all IPSec SAs. For more information, see the *Junos Network Management Configuration Guide*.

Option: Configuring IPSec Dynamic Endpoints

IPSec tunnels can also be established using *dynamic peer* security gateways, in which the remote end of the tunnels do not have a statically assigned IPv4 or IPv6 address. Since the remote address is not known and is assigned from an address pool each time the remote host reboots, establishment of the tunnel relies on using IKE main mode with preshared global keys. Both policy-based and link-type tunnels are supported as follows:

- Policy-based tunnels used shared mode.
- Link-type or routed tunnels use dedicated mode. Each tunnel allocates a service interface from a pool of interfaces configured for the dynamic peers. Routing protocols

can be configured to run on these service interfaces to learn routes over the IPSec tunnel that is used as a link.

This section includes the following topics:

Dynamic Endpoint Tunnel Architecture

When you configure dynamic endpoint tunnels, the following components are used:

Authentication Process

The remote dynamic peer initiates IKE and IPSec negotiations with the local (Juniper Networks) router. The local router uses a default set of authentication and encryption values to match the IPSec and IKE proposals sent by the remote peer to establish the SA. If any of the values match, the tunnel establishment process continues. The default values are shown in Table 5 on page 37.

Table 5: Default IKE and IPSec Proposals for Dynamic SA Negotiations

Statement Name	Values
Implicit IKE Proposal	
authentication-method	preshared keys
dh-group	group1, group2
authentication-algorithm	sha1, md5
encryption-algorithm	3des-cbc, des-cbc
lifetime-seconds	3600 seconds
Implicit IPSec Proposal	
protocol	esp, ah, bundle
authentication-algorithm	hmac-sha1-96, hmac-md5-96
encryption-algorithm	3des-cbc, des-cbc

Table 5: Default IKE and IPSec Proposals for Dynamic SA Negotiations *(continued)*

Statement Name	Values
lifetime-seconds	28,800 seconds (8 hours)

Phase 2 of the authentication process matches the *proxy identities* of the protected hosts and networks sent by the peer against a list of configured proxy identities. The accepted proxy identity is used to create the dynamic rules for encrypting the traffic. You can configure proxy identities by including the **allowed-proxy-pair** statement in an IKE access profile at the **[edit access profile *profile-name* client * ike]** hierarchy level. If no configured entry matches, the negotiation is rejected.

However, if you do not configure the **allowed-proxy-pair** statement, the default value **ANY(0.0.0.0/0)-ANY** is applied, and the local router accepts any proxy identities sent by the peer.

Once the phase 2 negotiation has been successfully completed, the router builds dynamic rules and inserts the reverse route into the routing table using the accepted proxy identity.

Dynamic Implicit Rules

After successful negotiation with the dynamic peer, the key management process (kmd) creates a dynamic rule for the accepted phase 2 proxy and applies it on the local AS or MultiServices PIC. The source and destination addresses are specified by the accepted proxy. This rule is used to encrypt traffic directed to one of the end hosts in the phase 2 proxy identity.

NOTE: You do not configure this rule; it is created by the key management process (kmd).

The **ipsec-inside-interface** value is the interface name assigned to the dynamic tunnel. The **source-address** and **destination-address** values are accepted from the proxy ID. The **match-direction** value is **input** for next-hop-style service sets.

Rule lookup for static tunnels is unaffected by the presence of a dynamic rule; it is performed in the order configured. When a packet is received for a service-set, static rules are always matched first. Dynamic rules are matched only after the rule match for static rules has failed.

Reverse Route Insertion

Static routes are automatically inserted into the route table for those networks and hosts protected by a remote tunnel endpoint. These protected hosts and networks are known as remote proxy identities.

Each route is created based on the remote proxy network and prefix length sent by the peer and is inserted in the relevant route table after successful phase 1 and phase 2 negotiations.

The route preference for each of these static reverse routes is 1. This value is necessary to avoid conflict with similar routes that might be added by the routing protocol process (rpd).

No routes are added if the accepted remote proxy address is the default (**0.0.0.0/0**). In this case, you can run routing protocols over the IPSec tunnel to learn routes and add static routes for the traffic you want to be protected over this tunnel.

For next-hop style service sets, the reverse routes include next hops pointing to the locations specified by the **inside-service-interface** statements.

The selection of the routing table in which these routes are inserted depends on where you configure the **inside-service-interface** statement. If these interfaces are present in a VRF routing instance, then routes are added to the corresponding VRF routing table; otherwise, the routes are added to **inet.0**.

NOTE: Reverse route insertion takes place only for tunnels to dynamic peers. These routes are added only for next-hop style service sets.

Configuring an IKE Access Profile

You can configure only one tunnel profile per service set for all dynamic peers. The configured preshared key in the profile is used for IKE authentication of all dynamic peers terminating in that service set.

The IKE tunnel profile specifies all the information needed to complete the IKE negotiation. For more information on access profiles, see the *Junos System Basics Configuration Guide*.

```
[edit access]
profile profile-name {
   client * {
      ike {
         allowed-proxy-pair {
            remote remote-proxy-address local local-proxy-address;
         }
         pre-shared-key ([ ascii-text key-string ] | [hexadecimal key-string ]);
         interface-id string-value;
         ipsec-policy ipsec-policy;
      }
   }
}
```

NOTE: For dynamic peers, the Junos OS supports only IKE main mode with the preshared key method of authentication. In this mode, an IPv4 or IPv6 address is used to identify a tunnel peer to get the preshared key information. The client value * (wildcard) means that the configuration within this profile is valid for all dynamic peers terminating within the service set accessing this profile.

The following statements are the parts of the IKE profile:

- **allowed-proxy-pair**—During phase 2 IKE negotiation, the remote peer supplies its network address (**remote**) and its peer's network address (**local**). Since multiple dynamic tunnels are authenticated through the same mechanism, this statement must include the list of possible combinations. If the dynamic peer does not present a valid combination, the phase 2 IKE negotiation fails.

  By default, remote 0.0.0.0/0 local 0.0.0.0/0 is used if no values are configured.

- **pre-shared-key**—Mandatory key used to authenticate the dynamic peer during IKE phase 1 negotiation. This key must be configured on both ends of the tunnel and distributed through an out-of-band secure mechanism. You can configure the key value either in **hexadecimal** or **ascii-text** format.

- **interface-id**—Interface identifier, a mandatory attribute used to derive the logical service interface information for the session.

- **ipsec-policy**—Name of the IPsec policy that defines the IPsec policy information for the session. You define the IPsec policy at the **[edit services ipsec-vpn ipsec policy *policy-name*]** hierarchy level. If no policy is set, any policy proposed by the dynamic peer is accepted.

Configuring the Service Set

To complete a dynamic endpoint tunnel configuration, you need to reference the IKE access profile configured at the [**edit access**] hierarchy level in the service set. To do this, include the **ike-access-profile** statement at the [**edit services service-set *name* ipsec-vpn-options**] hierarchy level:

```
[edit services]
service-set name {
    next-hop-service {
        inside-service-interface interface-name;
        outside-service-interface interface-name;
    }
    ipsec-vpn-options {
        local-gateway address;
        ike-access-profile profile-name;
    }
}
```

You can reference only one access profile in each service set. This profile is used to negotiate IKE and IPSec security associations with dynamic peers only.

NOTE: If you configure an IKE access profile in a service set, no other service set can share the same local-gateway address.

Configuring the Interface Identifier

You can configure an interface identifier for a group of dynamic peers, which specifies which adaptive services logical interface(s) take part in the dynamic IPSec negotiation. By assigning the same interface identifier to multiple logical interfaces, you can create a pool of interfaces for this purpose. To configure, include the **ipsec-interface-id** statement at the [**edit interfaces** *interface-name*] hierarchy level:

```
[edit interfaces sp-fpc/pic/port]
unit logical-unit-number {
  dial-options {
    ipsec-interface-id identifier;
    (shared | dedicated);
  }
}
```

Specifying the interface identifierin the **dial-options** statement makes this logical interface part of the pool identified by the IPSec interface identifier.

NOTE: Only one interface identifier can be specified at a time. You can include the **ipsec-interface-id** statement or the **l2tp-interface-id** statement, but not both simultaneously.

The **shared** statement enables one logical interface to be shared across multiple tunnels. The **dedicated** statement specifies that the logical interface is associated with a single tunnel, which is necessary when you are configuring an IPSec link-type tunnel. You must include the **dedicated** statement when you specify an **ipsec-interface-id** value.

Option: Configuring Multiple Routed Tunnels in a Single Next-Hop Service Set

To save you time and simplify your configurations, an enhancement to the Junos OS enables you to configure several routed IPSec tunnels within a single next-hop service set. To configure, establish multiple services interfaces as inside interfaces by including the **service-domain inside** statement at the **[edit interfaces sp-*fpc/pic/port* unit *logical-unit-number*]** hierarchy level. Then, include the **ipsec-inside-interface** statement at the **[edit services ipsec-vpn rule *rule-name* term *term-name* from]** hierarchy level.

NOTE: The full IPSec and IKE proposals and policies are not shown in the following example for the sake of brevity. For more information on proposals and policies, see "Configuring IKE Dynamic SAs" on page 20.

```
[edit]
interfaces {
```

```
    sp-3/3/0 {
        unit 3 {
            family inet;
            service-domain inside;
        }
        unit 4 {
            family inet;
            service-domain outside;
        }
        unit 5 {
            family inet;
            service-domain inside;
        }
    }
}
services {
    service-set link_type_ss_1 {
        next-hop-service {
            inside-service-interface sp-3/3/0.3;
            outside-service-interface sp-3/3/0.4;
        }
        ipsec-vpn-options {
            local-gateway 10.8.7.2;
        }
        ipsec-vpn-rules link_rule_1;
    }
    ipsec-vpn {
        rule link_rule_1 {
            term 1 {
                from {
                    ipsec-inside-interface sp-3/3/0.3;
                }
                then {
                    remote-gateway 10.10.7.3;
                    backup-remote-gateway 10.8.7.1;
                    dynamic {
                        ike-policy main_mode_ike_policy;
                        ipsec-policy dynamic_ipsec_policy;
                    }
                }
            }
            term 2 {
                from {
                    ipsec-inside-interface sp-3/3/0.5;
                }
                then {
                    remote-gateway 10.12.7.5;
                    dynamic {
                        ike-policy main_mode_ike_policy;
                        ipsec-policy dynamic_ipsec_policy;
                    }
                }
            }
            match-direction input;
        }
    }
```

```
}
```

To confirm that your configuration is working, issue the **show services ipsec-vpn ipsec security-associations** command. Notice that each IPSec inside interface that you assigned to each IPSec tunnel is included in the output of this command.

```
user@router> show services ipsec-vpn ipsec security-associations
Service set: link_type_ss_1

  Rule: link_rule_1, Term: 1, Tunnel index: 1
  Local gateway: 10.8.7.2, Remote gateway: 10.8.7.1
  IPSec inside interface: sp-3/3/0.3
    Direction SPI         AUX-SPI     Mode       Type     Protocol
    inbound   3216392497  0           tunnel     dynamic  ESP
    outbound  398917249   0           tunnel     dynamic  ESP

  Rule: link_rule_1, Term: 2, Tunnel index: 2
  Local gateway: 10.8.7.2, Remote gateway: 10.12.7.5
  IPSec inside interface: sp-3/3/0.5
    Direction SPI         AUX-SPI     Mode       Type     Protocol
    inbound   762146783   0           tunnel     dynamic  ESP
    outbound  319191515   0           tunnel     dynamic  ESP
```

IPSec Configuration Examples

This section contains configuration examples and commands you can use to verify your IPSec configuration:

- Example: ES PIC Manual SA Configuration on page 43
- Example: AS PIC Manual SA Configuration on page 52
- Example: ES PIC IKE Dynamic SA Configuration on page 60
- Example: AS PIC IKE Dynamic SA Configuration on page 71
- Example: IKE Dynamic SA Between an AS PIC and an ES PIC Configuration on page 79
- Example: AS PIC IKE Dynamic SA with Digital Certificates Configuration on page 90
- Example: Dynamic Endpoint Tunneling Configuration on page 108

Example: ES PIC Manual SA Configuration

Figure 3: ES PIC Manual SA Topology Diagram

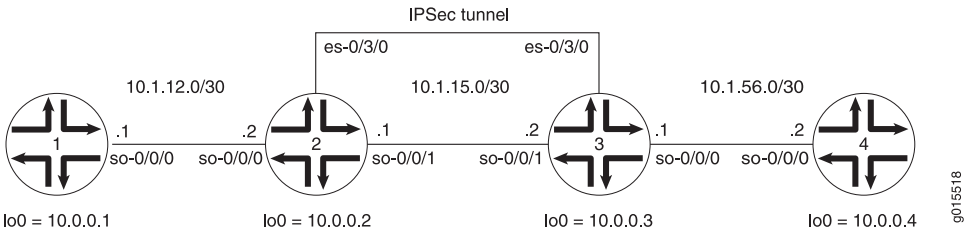


Figure 3 on page 43 shows an IPSec topology containing a group of four routers. Routers 2 and 3 establish an IPSec tunnel using an ES PIC and manual SA settings. Routers 1 and 4 provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R2 so-0/0/0";
      unit 0 {
         family inet {
            address 10.1.12.2/30;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.0.0.1/32;
         }
      }
   }
}
routing-options {
   router-id 10.0.0.1;
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface lo0.0;
      }
   }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional manual SA called **sa-manual** at the **[edit security ipsec security-association]** hierarchy level. Use AH for the protocol, **400** for the SPI, HMAC-MD5-96 for authentication, and a 32-bit hexadecimal authentication key for the MD5 authentication key. (For more information about key length, see Table 2 on page 17.) Because you are using AH, there is no need to configure encryption.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 1 destined for Router 4, whereas the **es-return** filter matches the return path from Router 4 to Router 1. Apply the **es-traffic** filter to the **so-0/0/0** interface; then apply both the **es-return** filter and the **sa-manual** SA to the **es-0/3/0** interface.

Router 2

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R1 so-0/0/0";
      unit 0 {
         family inet {
            filter {
               input es-traffic; # Apply a filter that sends traffic to the IPSec tunnel here.
            }
            address 10.1.12.1/30;
```

```
        }
      }
    }
    so-0/0/1 {
      description "To R3 so-0/0/1";
      unit 0 {
        family inet {
          address 10.1.15.1/30;
        }
      }
    }
    es-0/3/0 {
      unit 0 {
        tunnel { # Specify the IPSec tunnel endpoints here.
          source 10.1.15.1;
          destination 10.1.15.2;
        }
        family inet {
          ipsec-sa sa-manual; # Apply the manual SA here.
          filter {
            input es-return; # Apply the filter that matches return IPSec traffic here.
          }
        }
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.0.0.2/32;
        }
      }
    }
}
routing-options {
    router-id 10.0.0.2;
}
protocols {
    ospf {
      area 0.0.0.0 {
        interface so-0/0/0.0;
        interface so-0/0/1.0;
        interface lo0.0;
      }
    }
}
security {
    ipsec {
      security-association sa-manual { # Define the manual SA specifications here.
        mode tunnel;
        manual {
          direction bidirectional {
            protocol ah;
            spi 400;
            authentication {
              algorithm hmac-md5-96;
              key hexadecimal "$9$rO/eK8x7VY2ahSvL7-2gfTQF9Apu1EhrmfF/CtI
```

```
                    RlKMW7-VwYg4ZhSeW8XbwoJGjHmP5QF69wY4Zjif5369ApBSyKv8XRE";
                }
            }
        }
    }
}

# The 32-bit unencrypted hexadecimal key is abcdef01abcdef01abcdef01abcdef01.
firewall {
    filter es-traffic { # Define a filter that sends traffic to the IPSec tunnel here.
        term to-es {
            from {
                source-address {
                    10.1.12.0/24;
                }
                destination-address {
                    10.1.56.0/24;
                }
            }
            then {
                count ipsec-tunnel;
                ipsec-sa sa-manual;
            }
        }
        term other {
            then accept;
        }
    }
    filter es-return { # Define a filter that matches return IPSec traffic here.
        term return {
            from {
                source-address {
                    10.1.56.0/24;
                }
                destination-address {
                    10.1.12.0/24;
                }
            }
            then accept;
        }
    }
}
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional manual SA called **sa-manual** at the **[edit security ipsec security-association]** hierarchy level. Use the exact same specifications that you used for the SA on Router 2: AH for the protocol, **400** for the SPI, HMAC-MD5-96 for authentication, and a 32-bit hexadecimal authentication key of **abcdef01abcdef01abcdef01abcdef01** for the MD5 authentication key. (For more information about authentication key length, see Table 2 on page 17.) Because you are using AH, there is no need to configure an encryption algorithm.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 4 destined for Router 1, whereas the

es-return filter matches the return path from Router 1 to Router 4. Apply the **es-traffic** filter to the **so-0/0/0** interface; then apply both the **es-return** filter and the **sa-manual** SA to the **es-0/3/0** interface.

Router 3

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R4 so-0/0/0";
      unit 0 {
         family inet {
            filter {
               input es-traffic; # Apply a filter that sends traffic to the IPSec tunnel here.
            }
            address 10.1.56.1/30;
         }
      }
   }
   so-0/0/1 {
      description "To R2 so-0/0/1";
      unit 0 {
         family inet {
            address 10.1.15.2/30;
         }
      }
   }
   es-0/3/0 {
      unit 0 {
         tunnel { # Specify the IPSec tunnel endpoints here.
            source 10.1.15.2;
            destination 10.1.15.1;
         }
         family inet {
            ipsec-sa sa-manual; # Apply the manual SA here.
            filter {
               input es-return; # Apply the filter that matches return IPSec traffic here.
            }
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.0.0.3/32;
         }
      }
   }
}
routing-options {
   router-id 10.0.0.3;
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface so-0/0/1.0;
         interface lo0.0;
```

```
    }
  }
}
security {
  ipsec {
    security-association sa-manual { # Define the manual SA specifications here.
      mode tunnel;
      manual {
        direction bidirectional {
          protocol ah;
          spi 400;
          authentication {
            algorithm hmac-md5-96;
            key hexadecimal "$9$KMfMWx-ds4oGyl87dboaQF36tuOBESyK5Q6
              Ap0hcvWLXdbs24aJDylMXxNY2ZUjk.5Tz36Ct24JDkqQz/CtuORleW8xNcS";
          }
        }
      }
    }
  }
}


## The 32-bit unencrypted hexadecimal key is abcdef01abcdef01abcdef01abcdef01.
firewall {
  filter es-traffic { # Define a filter that sends traffic to the IPSec tunnel here.
    term to-es {
      from {
        source-address {
          10.1.56.0/24;
        }
        destination-address {
          10.1.12.0/24;
        }
      }
      then {
        count ipsec-tunnel;
        ipsec-sa sa-manual;
      }
    }
    term other {
      then accept;
    }
  }
  filter es-return { # Define a filter that matches return IPSec traffic here.
    term return {
      from {
        source-address {
          10.1.12.0/24;
        }
        destination-address {
          10.1.56.0/24;
        }
      }
      then accept;
```

```
    }
  }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R3 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.56.2/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.4/32;
      }
    }
  }
}
routing-options {
  router-id 10.0.0.4;
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface lo0.ping
    }
  }
}
```

Verifying Your Work

To verify proper operation of a manual IPSec SA on the ES PIC, use the following commands:

- **ping**
- **show ipsec security-associations (detail)**
- **traceroute**

The following sections show the output of these commands used with the configuration example:

- Router 1 on page 50
- Router 2 on page 50
- Router 3 on page 51
- Router 4 on page 51

Router 1

On Router 1, issue a **ping** command to the **so-0/0/0** interface of Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=253 time=0.939 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=253 time=0.886 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=253 time=0.826 ms
^C
--- 10.1.56.2 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.826/0.884/0.939/0.046 ms
```

You can also issue the **traceroute** command to verify that traffic to **10.1.56.2** travels over the IPSec tunnel between Router 2 and Router 3. Notice that the second hop does not reference **10.1.15.2**—the physical interface on Router 3. Instead, the loopback address of **10.0.0.3** on Router 3 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R1> traceroute 10.1.56.2
traceroute to 10.1.56.2 (10.1.56.2), 30 hops max, 40 byte packets
 1  10.1.12.1 (10.1.12.1)  0.655 ms  0.549 ms  0.508 ms
 2  10.0.0.3 (10.0.0.3)  0.833 ms  0.786 ms  0.757 ms
 3  10.1.56.2 (10.1.56.2)  0.808 ms  0.741 ms  0.716 ms
```

Router 2

Another way to verify that matched traffic is being diverted to the bidirectional IPSec tunnel is to view the firewall filter counter. After you issue the **ping** command from Router 1 (three packets), the **es-traffic** firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          252                    3
```

After you issue the **ping** command from both Router 1 (three packets) and Router 4 (two packets), the **es-traffic** firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          420                    5
```

To verify that the IPSec security association is active, issue the **show ipsec security-associations detail** command. Notice that the SA contains the settings you specified, such as AH for the protocol and HMAC-MD5-96 for the authentication algorithm.

```
user@R2> show ipsec security-associations detail
Security association: sa-manual, Interface family: Up

  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
```

```
  Direction: inbound, SPI: 400, AUX-SPI: 0
   Mode: tunnel, Type: manual, State: Installed
   Protocol: AH, Authentication: hmac-md5-96, Encryption: None
   Anti-replay service: Disabled

   Direction: outbound, SPI: 400, AUX-SPI: 0
   Mode: tunnel, Type: manual, State: Installed
   Protocol: AH, Authentication: hmac-md5-96, Encryption: None
   Anti-replay service: Disabled
```

Router 3

View the firewall filter counter to continue verifying that matched traffic is being diverted to the bidirectional IPSec tunnel. After you issue the **ping** command from Router 1 (three packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          252                    3
```

After you issue the **ping** command from both Router 1 (three packets) and Router 4 (two packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          420                    5
```

To verify that the IPSec security association is active, issue the **show ipsec security-associations detail** command. Notice that the SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ipsec security-associations detail
Security association: sa-manual, Interface family: Up

  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 400, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: AH, Authentication: hmac-md5-96, Encryption: None
    Anti-replay service: Disabled

    Direction: outbound, SPI: 400, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: AH, Authentication: hmac-md5-96, Encryption: None
    Anti-replay service: Disabled
```

Router 4

On Router 4, issue a **ping** command to the **so-0/0/0** interface of Router 1 to send traffic across the IPSec tunnel.

```
user@R4> ping 10.1.12.2
```

```
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=253 time=0.937 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=253 time=0.872 ms
^C
--- 10.1.12.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.872/0.905/0.937/0.032 ms
```

You can also issue the **traceroute** command to verify that traffic to **10.1.12.2** travels over the IPSec tunnel between Router 3 and Router 2. Notice that the second hop does not reference **10.1.15.1**—the physical interface on Router 2. Instead, the loopback address of **10.0.0.2** on Router 2 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.56.1 (10.1.56.1)  0.670 ms  0.589 ms  0.548 ms
 2  10.0.0.2 (10.0.0.2)  0.815 ms  0.791 ms  0.763 ms
 3  10.1.12.2 (10.1.12.2)  0.798 ms  0.741 ms  0.714 ms
```

Example: AS PIC Manual SA Configuration

Figure 4: AS PIC Manual SA Topology Diagram

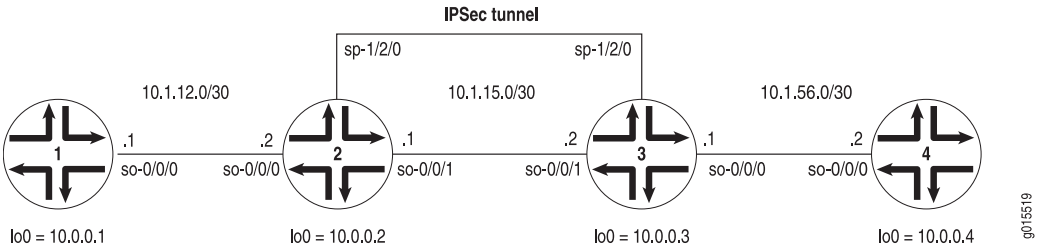


Figure 4 on page 52 shows a similar IPSec topology to the one used in the ES PIC manual SA example. The difference is that Routers 2 and 3 establish an IPSec tunnel using an AS PIC and use slightly modified manual SA settings. Routers 1 and 4 again provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
```

```
    }
  }
  routing-options {
    router-id 10.0.0.1;
  }
  protocols {
    ospf {
      area 0.0.0.0 {
        interface so-0/0/0.0;
        interface lo0.0;
      }
    }
  }
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional manual SA in a rule called rule-manual-SA-BiEspshades at the **[edit ipsec-vpn rule]** hierarchy level. Reference this rule in a service set called service-set-manual-BiEspshades at the **[edit services service-set]** hierarchy level.

Configure all specifications for your manual SA. Use ESP for the protocol, **261** for the SPI, HMAC-SHA1-96 for authentication, DES-CBC for encryption, a 20-bit ASCII authentication key for the SHA-1 authentication key, and an 8-bit ASCII encryption key for the DES-CBC authentication key. (For more information about key lengths, see Table 2 on page 17.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 2

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R1 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.12.1/30;
      }
    }
  }
  so-0/0/1 {
    description "To R3 so-0/0/1";
    unit 0 {
      family inet {
        address 10.1.15.1/30;
      }
    }
  }
  sp-1/2/0 {
    services-options {
      syslog {
        host local {
          services info;
        }
      }
    }
    unit 0 {
```

```
    family inet {
    }
    unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
      family inet;
      service-domain inside;
    }
    unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
      family inet;
      service-domain outside;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.2/32;
      }
    }
  }
}
routing-options {
  router-id 10.0.0.2;
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface lo0.0;
      interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
    }
  }
}
services {
  service-set service-set-manual-BiEspshades { # Define your service set here.
    next-hop-service { # Required for dynamic routing protocols such as OSPF.
      inside-service-interface sp-1/2/0.1;
      outside-service-interface sp-1/2/0.2;
    }
    ipsec-vpn-options {
      local-gateway 10.1.15.1; # Specify the local IP address of the IPSec tunnel.
    }
    ipsec-vpn-rules rule-manual-SA-BiEspshades; # Reference the IPSec rule here.
  }
  ipsec-vpn {
    rule rule-manual-SA-BiEspshades { # Define your IPSec VPN rule here.
      term term-manual-SA-BiEspshades {
        then {
          remote-gateway 10.1.15.2; # The remote IP address of the IPSec tunnel.
          manual { # Define the manual SA specifications here.
            direction bidirectional {
              protocol esp;
              spi 261;
              authentication {
                algorithm hmac-sha1-96;
                key ascii-text "$9$v.s8xd24Zk.5bs.5QFAtM8XNVYJGifT3goT369
                  OBxNdw2ajHmFnCZUnCtuEh";
```

```
                    ## The unencrypted key is juniperjuniperjunipe (20 characters for
                      HMAC-SHA-1-96).
                  }
                  encryption {
                    algorithm des-cbc;
                    key ascii-text "$9$3LJW/A0EclLxdBIxdbsJZn/CpOR";
                    ## The unencrypted key is juniperj (8 characters for DES-CBC).
                  }
                }
              }
            }
          }
          match-direction input; # Correct match direction for next-hop service sets.
        }
      }
    }
    }
  security {
    pki {
      auto-re-enrollment {
        certificate-id certificate-name {
          ca-profile ca-profile-name;
          challenge-password password;
          re-enroll-trigger-time-percentage percentage; #Percentage of validity-period
            # (specified in certificate) when automatic
            # reenrollment should be initiated.
          re-generate-keypair;
          validity-period number-of-days;
        }
      }
    }
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional manual SA in a rule called rule-manual-SA-BiEspshades at the **[edit ipsec-vpn rule]** hierarchy level. Reference this rule in a service set called service-set-manual-BiEspshades at the **[edit services service-set]** hierarchy level.

Configure the same specifications for your manual SA that you specified on Router 2. Use ESP for the protocol, **261** for the SPI, HMAC-SHA1-96 for authentication, DES-CBC for encryption, a 20-bit ASCII authentication key for the SHA-1 authentication key, and an 8-bit ASCII encryption key for the DES-CBC authentication key. (For more information about key lengths, see Table 2 on page 17.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 3

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R4 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.56.1/30;
      }
```

```
    }
  }
  so-0/0/1 {
    description "To R2 so-0/0/1";
    unit 0 {
      family inet {
        address 10.1.15.2/30;
      }
    }
  }
  sp-1/2/0 {
    services-options {
      syslog {
        host local {
          services info;
        }
      }
    }
    unit 0 {
      family inet {
      }
      unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
        family inet;
        service-domain inside;
      }
      unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
        family inet;
        service-domain outside;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.0.0.3/32;
        }
      }
    }
  }
  routing-options {
    router-id 10.0.0.3;
  }
  protocols {
    ospf {
      area 0.0.0.0 {
        interface so-0/0/0.0;
        interface lo0.0;
        interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
      }
    }
  }
  services {
    service-set service-set-manual-BiEspshades { # Define your service set here.
      next-hop-service { # Required for dynamic routing protocols such as OSPF.
        inside-service-interface sp-1/2/0.1;
        outside-service-interface sp-1/2/0.2;
      }
```

```
        ipsec-vpn-options {
          local-gateway 10.1.15.2; # Specify the local IP address of the IPSec tunnel.
        }
        ipsec-vpn-rules rule-manual-SA-BiEspshades; # Reference the IPSec rule here.
      }
      ipsec-vpn {
        rule rule-manual-SA-BiEspshades { # Define your IPSec VPN rule here.
          term term-manual-SA-BiEspshades {
            then {
              remote-gateway 10.1.15.1; # The remote IP address of the IPSec tunnel.
              manual { # Define the manual SA specifications here.
                direction bidirectional {
                  protocol esp;
                  spi 261;
                  authentication {
                    algorithm hmac-sha1-96;
                    key ascii-text "$9$v.s8xd24Zk.5bs.5QFAtM8XNVYJGifT3goT369
                      OBxNdw2ajHmFnCZUnCtuEh";
                    ## The unencrypted key is juniperjuniperjunipe (20 characters for
                      HMAC-SHA-1-96).
                  }
                  encryption {
                    algorithm des-cbc;
                    key ascii-text "$9$3LJW/A0EclLxdBIxdbsJZn/CpOR";
                    ## The unencrypted key is juniperj (8 characters for DES-CBC).
                  }
                }
              }
            }
          }
          match-direction input; # Specify in which direction the rule should match.
        }
      }
    }
  }
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R3 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.56.2/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.4/32;
      }
    }
  }
}
```

```
routing-options {
   router-id 10.0.0.4;
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface lo0.0;
      }
   }
}
```

Verifying Your Work

To verify proper operation of a manual IPSec SA on the AS PIC, use the following commands:

- **ping**
- **show services ipsec-vpn ipsec security-associations** (detail)
- **show services ipsec-vpn ipsec statistics**

The following sections show the output of these commands used with the configuration example:

- Router 1 on page 58
- Router 2 on page 58
- Router 3 on page 59

Router 1

On Router 1, issue a **ping** command to the **lo0** interface on Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4): 56 data bytes
64 bytes from 10.0.0.4: icmp_seq=0 ttl=254 time=1.375 ms
64 bytes from 10.0.0.4: icmp_seq=1 ttl=254 time=18.375 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=254 time=1.120 ms
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.120/6.957/18.375/8.075 ms
```

Router 2

To verify that the IPSec security association is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. Notice that the SA contains the settings you specified, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-manual-BiEspshades
  Rule: rule-manual-SA-BiEspshades, Term: term-manual-SA-BiEspshades,
  Tunnel index: 1
```

```
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.0.0.0/8)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled

    Direction: outbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled
```

To verify that traffic is traveling over the bidirectional IPSec tunnel, issue the **show services ipsec-vpn statistics** command:

```
user@R2> show services ipsec-vpn ipsec statistics

PIC: sp-1/2/0, Service set: service-set-manual-BiEspshades

ESP Statistics:
  Encrypted bytes:             1616
  Decrypted bytes:             1560
  Encrypted packets:             20
  Decrypted packets:             19
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

Router 3

To verify that the IPSec security association is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-manual-BiEspshades
  Rule: rule-manual-SA-BiEspshades, Term: term-manual-SA-BiEspshades,
  Tunnel index: 1
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.0.0.0/8)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled

    Direction: outbound, SPI: 261, AUX-SPI: 0
    Mode: tunnel, Type: manual, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: des-cbc
    Anti-replay service: Disabled
```

To verify that traffic is traveling over the bidirectional IPSec tunnel, issue the **show services ipsec-vpn statistics** command:

```
user@R3> show services ipsec-vpn ipsec statistics
PIC: sp-1/2/0, Service set: service-set-manual-BiEspshades
ESP Statistics:
  Encrypted bytes:             1560
  Decrypted bytes:             1616
  Encrypted packets:             19
  Decrypted packets:             20
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

Example: ES PIC IKE Dynamic SA Configuration

Figure 5: ES PIC IKE Dynamic SA Topology Diagram

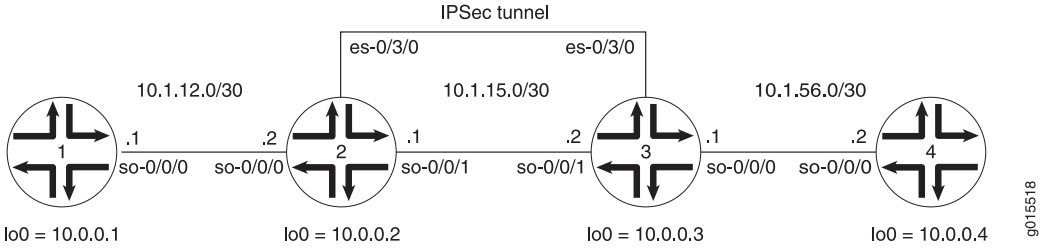


Figure 5 on page 60 shows the same IPSec topology as seen in the ES PIC manual SA example. However, this time the configuration requires Routers 2 and 3 to establish an IPSec tunnel using an IKE dynamic SA, enhanced authentication, and stronger encryption. Routers 1 and 4 continue to provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R2 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.12.2/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.1/32;
      }
    }
```

```
    }
  }
  routing-options {
    router-id 10.0.0.1;
  }
  protocols {
    ospf {
      area 0.0.0.0 {
        interface so-0/0/0.0;
        interface lo0.0;
      }
    }
  }
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA called **sa-dynamic** at the **[edit security ipsec security-association]** hierarchy level. For your IKE policy and proposal, use preshared keys for the authentication method, SHA-1 for the authentication algorithm, 3DES-CBC for encryption, group 2 for the Diffie-Hellman group, main mode, 3600 seconds for the lifetime, and a preshared key of **juniper** for the initial IKE negotiation. For your IPSec policy and proposal, use ESP for the protocol, HMAC-SHA1-96 for authentication, 3DES-CBC for encryption, 28800 seconds for the lifetime, and group 2 for the PFS group.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 1 destined for Router 4, whereas the **es-return** filter matches the return path from Router 4 to Router 1. Apply the **es-traffic** filter to the **so-0/0/0** interface, and then apply both the **es-return** filter and the **sa-dynamic** SA to the **es-0/3/0** interface.

Router 2

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R1 so-0/0/0";
    unit 0 {
      family inet {
        filter {
          input es-traffic; # Apply a filter that sends traffic to the IPSec tunnel here.
        }
        address 10.1.12.1/30;
      }
    }
  }
  so-0/0/1 {
    description "To R3 so-0/0/1";
    unit 0 {
      family inet {
        address 10.1.15.1/30;
      }
    }
  }
  es-0/3/0 {
    unit 0 {
      tunnel { # Specify the IPSec tunnel endpoints here.
        source 10.1.15.1;
        destination 10.1.15.2;
```

```
        }
        family inet {
          ipsec-sa sa-dynamic; # Apply the dynamic SA here.
          filter {
            input es-return; # Apply the filter that matches return IPSec traffic here.
          }
        }
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.0.0.2/32;
        }
      }
    }
}
routing-options {
  router-id 10.0.0.2;
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface so-0/0/1.0;
      interface lo0.0;
    }
  }
}
security {
  ipsec {
    proposal es-ipsec-proposal { # Define your IPSec proposal specifications here.
      protocol esp;
      authentication-algorithm hmac-sha1-96;
      encryption-algorithm 3des-cbc;
      lifetime-seconds 28800;
    }
    policy es-ipsec-policy { # Define your IPSec policy specifications here.
      perfect-forward-secrecy {
        keys group2;
      }
      proposals es-ipsec-proposal; # Reference the IPSec proposal here.
    }
    security-association sa-dynamic { # Define your dynamic SA here.
      mode tunnel;
      dynamic {
        ipsec-policy es-ipsec-policy; # Reference the IPSec policy here.
      }
    }
  }
  ike {
    proposal es-ike-proposal { # Define your IKE proposal specifications here.
      authentication-method pre-shared-keys;
      dh-group group2;
      authentication-algorithm sha1;
      encryption-algorithm 3des-cbc;
```

```
        lifetime-seconds 3600;
      }
      policy 10.1.15.2 { # Define your IKE policy specifications here.
        mode main;
        proposals es-ike-proposal; # Reference the IKE proposal here.
        pre-shared-key ascii-text "$9$TF6ABIcvWxp0WxNdg4QFn";
        ## The unencrypted preshared key for this example is juniper.
      }
    }
  }
  firewall {
    filter es-traffic { # Define a filter that sends traffic to the IPSec tunnel here.
      term to-es {
        from {
          source-address {
            10.1.12.0/24;
          }
          destination-address {
            10.1.56.0/24;
          }
        }
        then {
          count ipsec-tunnel;
          ipsec-sa sa-dynamic;
        }
      }
      term other {
        then accept;
      }
    }
    filter es-return { # Define a filter that matches return IPSec traffic here.
      term return {
        from {
          source-address {
            10.1.56.0/24;
          }
          destination-address {
            10.1.12.0/24;
          }
        }
        then accept;
      }
    }
  }
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA called **sa-dynamic** at the **[edit security ipsec security-association]** hierarchy level. Use the same policies and proposals that you used on Router 2.

For your IKE policy and proposal, use preshared keys for the authentication method, SHA-1 for the authentication algorithm, 3DES-CBC for encryption, group 2 for the Diffie-Hellman group, main mode, 3600 seconds for the lifetime, and a preshared key of **juniper** for the initial IKE negotiation. For your IPSec policy and proposal, use ESP for

the protocol, HMAC-SHA1-96 for authentication, 3DES-CBC for encryption, 28800 seconds for the lifetime, and group 2 for the PFS group.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 4 destined for Router 1, whereas the **es-return** filter matches the return path from Router 1 to Router 4. Apply the **es-traffic** filter to the **so-0/0/0** interface; then apply both the **es-return** filter and the **sa-dynamic** SA to the **es-0/3/0** interface.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R4 so-0/0/0";
        unit 0 {
            family inet {
                filter {
                    input es-traffic; # Apply a filter that sends traffic to the IPSec tunnel here.
                }
                address 10.1.56.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R2 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    es-0/3/0 {
        unit 0 {
            tunnel { # Specify the IPSec tunnel endpoints here.
                source 10.1.15.2;
                destination 10.1.15.1;
            }
            family inet {
                ipsec-sa sa-dynamic; # Apply the dynamic SA here.
                filter {
                    input es-return; # Apply the filter that matches return IPSec traffic here.
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.3/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.3;
}
```

```
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface so-0/0/1.0;
      interface lo0.0;
    }
  }
}
security {
  ipsec {
    proposal es-ipsec-proposal { # Define your IPSec proposal specifications here.
      protocol esp;
      authentication-algorithm hmac-sha1-96;
      encryption-algorithm 3des-cbc;
      lifetime-seconds 28800;
    }
    policy es-ipsec-policy { # Define your IPSec policy specifications here.
      perfect-forward-secrecy {
        keys group2;
      }
      proposals es-ipsec-proposal; # Reference the IPSec proposal here.
    }
    security-association sa-dynamic { # Define your dynamic SA here.
      mode tunnel;
      dynamic {
        ipsec-policy es-ipsec-policy; # Reference the IPSec policy here.
      }
    }
  }
  ike {
    proposal es-ike-proposal { # Define your IKE proposal specifications here.
      authentication-method pre-shared-keys;
      dh-group group2;
      authentication-algorithm sha1;
      encryption-algorithm 3des-cbc;
      lifetime-seconds 3600;
    }
    policy 10.1.15.1 { # Define your IKE policy specifications here.
      mode main;
      proposals es-ike-proposal; # Reference the IKE proposal here.
      pre-shared-key ascii-text "$9$TF6ABIcvWxp0WxNdg4QFn";
      ## The unencrypted preshared key for this example is juniper.
    }
  }
}
firewall {
  filter es-traffic { # Define a filter that sends traffic to the IPSec tunnel here.
    term to-es {
      from {
        source-address {
          10.1.56.0/24;
        }
        destination-address {
          10.1.12.0/24;
        }
```

```
        }
        then {
          count ipsec-tunnel;
          ipsec-sa sa-dynamic;
        }
      }
      term other {
        then accept;
      }
    }
    filter es-return { # Define a filter that matches return IPSec traffic here.
      term return {
        from {
          source-address {
            10.1.12.0/24;
          }
          destination-address {
            10.1.56.0/24;
          }
        }
        then accept;
      }
    }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R3 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.56.2/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.4/32;
      }
    }
  }
}
routing-options {
  router-id 10.0.0.4;
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface lo0.0;
    }
  }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the ES PIC, use the following commands:

- **ping**
- **show ike security-associations (detail)**
- **show ipsec security-associations (detail)**
- **traceroute**

The following sections show the output of these commands used with the configuration example:

- Router 1 on page 67
- Router 2 on page 68
- Router 3 on page 69
- Router 4 on page 70

Router 1

On Router 1, issue a **ping** command to the **so-0/0/0** interface of Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=253 time=0.917 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=253 time=0.881 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=253 time=0.897 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=253 time=0.871 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=253 time=0.890 ms
64 bytes from 10.1.56.2: icmp_seq=5 ttl=253 time=0.858 ms
64 bytes from 10.1.56.2: icmp_seq=6 ttl=253 time=0.904 ms
^C
--- 10.1.56.2 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.858/0.888/0.917/0.019 ms
```

You can also issue the **traceroute** command to verify that traffic to **10.1.56.2** travels over the IPSec tunnel between Router 2 and Router 3. Notice that the second hop does not reference **10.1.15.2**—the physical interface on Router 3. Instead, the loopback address of **10.0.0.3** on Router 3 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R1> traceroute 10.1.56.2
traceroute to 10.1.56.2 (10.1.56.2), 30 hops max, 40 byte packets
 1  10.1.12.1 (10.1.12.1)  0.655 ms  0.549 ms  0.508 ms
 2  10.0.0.3 (10.0.0.3)  0.833 ms  0.786 ms  0.757 ms
```

3 10.1.56.2 (10.1.56.2) 0.808 ms 0.741 ms 0.716 ms

Router 2

Another way to verify that matched traffic is being diverted to the bidirectional IPSec tunnel is to view the firewall filter counter. After you issue the **ping** command from Router 1 (seven packets), the **es-traffic** firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          588                    7
```

After you issue the **ping** command from both Router 1 (seven packets) and Router 4 (five packets), the **es-traffic** firewall filter counter looks like this:

```
user@R2> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                         1008                   12
```

To verify that the IKE SA negotiation between Routers 2 and 3 is successful, issue the **show ike security-associations detail** command. Notice that the SA contains the settings you specified, such as SHA-1 for the authentication algorithm and 3DES-CBC for the encryption algorithm.

```
user@R2> show ike security-associations detail
IKE peer 10.1.15.2
  Role: Initiator, State: Matured
  Initiator cookie: b5dbdfe2f9000000, Responder cookie: a24c868410000041
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.1:500, Remote: 10.1.15.2:500
  Lifetime: Expires in 401 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                 1736
   Output bytes  :                 2652
   Input  packets:                    9
   Output packets:                   15
  Flags: Caller notification sent
  IPSec security associations: 3 created, 0 deleted
  Phase 2 negotiations in progress: 0
```

To verify that the IPSec security association is active, issue the **show ipsec security-associations detail** command. Notice that the SA contains the settings you specified, such as ESP for the protocol, HMAC-SHA1-96 for the authentication algorithm, and 3DES-CBC for the encryption algorithm.

```
user@R2> show ipsec security-associations detail
Security association: sa-dynamic, Interface family: Up
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
    Direction: inbound, SPI: 2133029543, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
```

```
    Soft lifetime: Expires in 26212 seconds
    Hard lifetime: Expires in 26347 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 1759450863, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26212 seconds
    Hard lifetime: Expires in 26347 seconds
    Anti-replay service: Disabled
```

Router 3

View the firewall filter counter to continue verifying that matched traffic is being diverted to the bidirectional IPSec tunnel. After you issue the **ping** command from Router 1 (seven packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          588                    7
```

After you issue the **ping** command from both Router 1 (seven packets) and Router 4 (five packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                         1008                   12
```

To verify the success of the IKE security association, issue the **show ike security-associations detail** command. Notice that the SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ike security-associations detail
IKE peer 10.1.15.1
  Role: Responder, State: Matured
  Initiator cookie: b5dbdfe2f9000000, Responder cookie: a24c868410000041
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.2:500, Remote: 10.1.15.1:500
  Lifetime: Expires in 564 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                 2652
   Output bytes  :                 1856
   Input  packets:                   15
   Output packets:                   10
  Flags: Caller notification sent
  IPSec security associations: 3 created, 4 deleted
  Phase 2 negotiations in progress: 0
```

To verify that the IPSec security association is active, issue the **show ipsec security-associations detail** command. Notice that the SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ipsec security-associations detail
```

```
Security association: sa-dynamic, Interface family: Up
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
    Direction: inbound, SPI: 1759450863, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26427 seconds
    Hard lifetime: Expires in 26517 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 2133029543, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26427 seconds
    Hard lifetime: Expires in 26517 seconds
    Anti-replay service: Disabled
```

Router 4

On Router 4, issue a **ping** command to the **so-0/0/0** interface of Router 1 to send traffic across the IPSec tunnel.

```
user@R4> ping 10.1.12.2
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=253 time=13.528 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=253 time=0.873 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=253 time=32.145 ms
64 bytes from 10.1.12.2: icmp_seq=3 ttl=253 time=0.921 ms
64 bytes from 10.1.12.2: icmp_seq=4 ttl=253 time=0.899 ms
^C
--- 10.1.12.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.873/9.673/32.145/12.255 ms
```

You can also issue the **traceroute** command to verify that traffic to **10.1.12.2** travels over the IPSec tunnel between Router 3 and Router 2. Notice that the second hop does not reference **10.1.15.1**—the physical interface on Router 2. Instead, the loopback address of **10.0.0.2** on Router 2 appears as the second hop. This indicates that the IPSec tunnel is operating correctly.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.56.1 (10.1.56.1)  0.681 ms  0.624 ms  0.547 ms
 2  10.0.0.2 (10.0.0.2)  0.800 ms  0.770 ms  0.737 ms
 3  10.1.12.2 (10.1.12.2)  0.793 ms  0.742 ms  0.716 ms
```

Example: AS PIC IKE Dynamic SA Configuration

Figure 6: AS PIC IKE Dynamic SA Topology Diagram

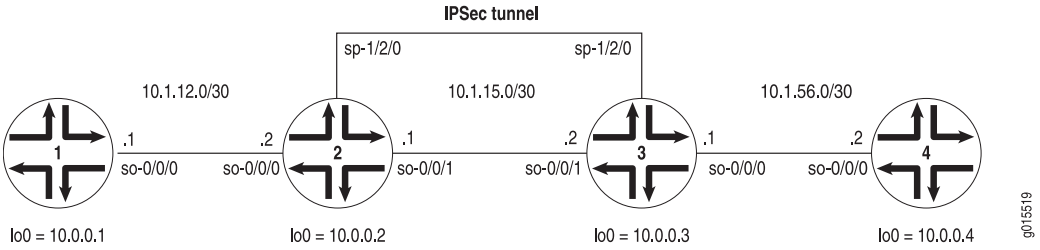


Figure 6 on page 71 shows the same IPSec topology as seen in the AS PIC manual SA example. However, this configuration requires Routers 2 and 3 to establish an IPSec tunnel using an IKE dynamic SA, enhanced authentication, and stronger encryption. Routers 1 and 4 continue to provide basic connectivity and are used to verify that the IPSec tunnel is operational.

NOTE: When you do not specify an IKE proposal, an IPSec proposal, and an IPSec policy on an AS PIC, the Junos OS defaults to the highest level of encryption and authentication. As a result, the default authentication protocol is ESP, the default authentication mode is HMAC-SHA1-96, and the default encryption mode is 3DES-CBC. For more information about default IKE and IPSec policies and proposals on the AS PIC, see "IKE and IPSec Proposal and Policy Default Values for the AS and MultiServices PICs" on page 20.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
```

```
        area 0.0.0.0 {
          interface so-0/0/0.0;
          interface lo0.0;
        }
      }
    }
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA in a rule called **rule-ike** at the **[edit ipsec-vpn rule]** hierarchy level. Reference this rule in a service set called **service-set-dynamic-BiEspsha3des** at the **[edit services service-set]** hierarchy level.

Using default values in the AS PIC, you do not need to specify an IPSec proposal, IPSec policy, or IKE proposal. However, you do need to configure a preshared key in an IKE policy with the **pre-shared-key** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see "IKE and IPSec Proposal and Policy Default Values for the AS and MultiServices PICs" on page 20.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 2

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R1 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.12.1/30;
      }
    }
  }
  so-0/0/1 {
    description "To R3 so-0/0/1";
    unit 0 {
      family inet {
        address 10.1.15.1/30;
      }
    }
  }
  sp-1/2/0 {
    services-options {
      syslog {
        host local {
          services info;
        }
      }
    }
    unit 0 {
      family inet {
      }
      unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
        family inet;
        service-domain inside;
```

```
    }
    unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
      family inet;
      service-domain outside;
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.2/32;
      }
    }
  }
}
routing-options {
  router-id 10.0.0.2;
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface lo0.0;
      interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
    }
  }
}
services {
  service-set service-set-dynamic-BiEspsha3des { # Define your service set here.
    next-hop-service { # Required for dynamic routing protocols such as OSPF.
      inside-service-interface sp-1/2/0.1;
      outside-service-interface sp-1/2/0.2;
    }
    ipsec-vpn-options {
      local-gateway 10.1.15.1; # Specify the local IP address of the IPSec tunnel.
    }
    ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
  }
  ipsec-vpn {
    rule rule-ike { # Define your IPSec VPN rule here.
      term term-ike {
        then {
          remote-gateway 10.1.15.2; # The remote IP address of the IPSec tunnel.
          dynamic { # This creates a dynamic SA.
            ike-policy ike-policy-preshared; # Reference your IKE policy here.
          }
        }
      }
      match-direction input; # Specify in which direction the rule should match.
    }
    ike {
      policy ike-policy-preshared { # Define your IKE policy specifications here.
        pre-shared-key ascii-text "$9$KtKWX-YgJHqfVwqfTzCAvWL";
        ## The unencrypted preshared key for this example is juniper.
      }
    }
  }
```

```
    }
  }
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA in a rule called **rule-ike** at the **[edit ipsec-vpn rule]** hierarchy level. Reference this rule in a service set called **service-set-dynamic-BiEspsha3des** at the **[edit services service-set]** hierarchy level.

Again, use the same default policies and proposals that you used on Router 2. However, remember to configure a preshared key in an IKE policy with the **pre-shared-key** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level. The key must match the one you specified on Router 2. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see "IKE and IPSec Proposal and Policy Default Values for the AS and MultiServices PICs" on page 20.)

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 3

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R4 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.56.1/30;
      }
    }
  }
  so-0/0/1 {
    description "To R2 so-0/0/1";
    unit 0 {
      family inet {
        address 10.1.15.2/30;
      }
    }
  }
  sp-1/2/0 {
    services-options {
      syslog {
        host local {
          services info;
        }
      }
    }
    unit 0 {
      family inet {
      }
      unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
        family inet;
        service-domain inside;
      }
      unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
        family inet;
        service-domain outside;
```

```
        }
      }
      lo0 {
        unit 0 {
          family inet {
            address 10.0.0.3/32;
          }
        }
      }
    }
    routing-options {
      router-id 10.0.0.3;
    }
    protocols {
      ospf {
        area 0.0.0.0 {
          interface so-0/0/0.0;
          interface lo0.0;
          interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
        }
      }
    }
    services {
      service-set service-set-dynamic-BiEspsha3des { # Define your service set here.
        next-hop-service { # Required for dynamic routing protocols such as OSPF.
          inside-service-interface sp-1/2/0.1;
          outside-service-interface sp-1/2/0.2;
        }
        ipsec-vpn-options {
          local-gateway 10.1.15.2; # Specify the local IP address of the IPSec tunnel.
        }
        ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
      }
      ipsec-vpn {
        rule rule-ike { # Define your IPSec VPN rule here.
          term term-ike {
            then {
              remote-gateway 10.1.15.1; # The remote IP address of the IPSec tunnel.
              dynamic { # This creates a dynamic SA.
                ike-policy ike-policy-preshared; # Reference your IKE policy here.
              }
            }
          }
          match-direction input; # Specify in which direction the rule should match.
        }
        ike {
          policy ike-policy-preshared { # Define your IKE policy specifications here.
            pre-shared-key ascii-text "$9$KtKWX-YgJHqfVwqfTzCAvWL";
            ## The unencrypted preshared key for this example is juniper.
          }
        }
      }
    }
  }
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R3 so-0/0/0";
      unit 0 {
         family inet {
            address 10.1.56.2/30;
         }
      }
   }
   lo0 {
      unit 0 {
         family inet {
            address 10.0.0.4/32;
         }
      }
   }
}
routing-options {
   router-id 10.0.0.4;
}
protocols {
   ospf {
      area 0.0.0.0 {
         interface so-0/0/0.0;
         interface lo0.0;
      }
   }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the AS PIC, use the following commands:

- **ping**
- **show services ipsec-vpn ike security-associations (detail)**
- **show services ipsec-vpn ipsec security-associations (detail)**
- **show services ipsec-vpn ipsec statistics**
- **traceroute**

The following sections show the output of these commands used with the configuration example:

- Router 1 on page 77
- Router 2 on page 77
- Router 3 on page 78
- Router 4 on page 79

Router 1

On Router 1, issue a **ping** command to the **so-0/0/0** interface on Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=254 time=1.351 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=254 time=1.187 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=254 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=254 time=1.154 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=254 time=1.156 ms
^C
--- 10.1.56.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.154/1.204/1.351/0.074 ms
```

Router 2

To verify that the IKE SA negotiation is successful, issue the **show services ipsec-vpn ike security-associations** command.

```
user@R2> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.2       Matured       03075bd3a0000003  4bff26a5c7000003  Main
```

To verify that the IPSec security association is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. Notice that the SA contains the default settings inherent in the AS PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des
  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
    Direction: inbound, SPI: 2666326758, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26863 seconds
    Hard lifetime: Expires in 26998 seconds
    Anti-replay service: Enabled, Replay window size: 64
    Direction: outbound, SPI: 684772754, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26863 seconds
    Hard lifetime: Expires in 26998 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To verify that traffic is traveling over the bidirectional IPSec tunnel, issue the **show services ipsec-vpn statistics** command:

```
user@R2> show services ipsec-vpn ipsec statistics
PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des
ESP Statistics:
  Encrypted bytes:             2248
  Decrypted bytes:             2120
  Encrypted packets:             27
```

```
  Decrypted packets:             25
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
```

Bad headers: 0, Bad trailers: 0

Router 3

To verify that the IKE SA negotiation is successful, issue the **show services ipsec-vpn ike security-associations** command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.1       Matured       03075bd3a0000003  4bff26a5c7000003  Main
```

To verify that the IPSec SA is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des
  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
    Direction: inbound, SPI: 684772754, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26598 seconds
    Hard lifetime: Expires in 26688 seconds
    Anti-replay service: Enabled, Replay window size: 64
    Direction: outbound, SPI: 2666326758, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 26598 seconds
    Hard lifetime: Expires in 26688 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To verify that traffic is traveling over the bidirectional IPSec tunnel, issue the **show services ipsec-vpn statistics** command:

```
user@R3> show services ipsec-vpn ipsec statistics
PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des
ESP Statistics:
  Encrypted bytes:            2120
  Decrypted bytes:            2248
  Encrypted packets:            25
  Decrypted packets:            27
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
```

```
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
```

Bad headers: 0, Bad trailers: 0

Router 4

On Router 4, issue a **ping** command to the **so-0/0/0** interface on Router 1 to send traffic across the IPSec tunnel.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=3 ttl=254 time=1.142 ms
64 bytes from 10.1.12.2: icmp_seq=4 ttl=254 time=1.139 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

The final way you can confirm that traffic travels over the IPSec tunnel is by issuing the **traceroute** command to the **so-0/0/0** interface on Router 1. Notice that the physical interface between Routers 2 and 3 is not referenced in the path; traffic enters the IPSec tunnel through the adaptive services IPSec inside interface on Router 3, passes through the loopback interface on Router 2, and ends at the **so-0/0/0** interface on Router 1.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.15.2 (10.1.15.2)  0.987 ms  0.630 ms  0.563 ms
 2  10.0.0.2 (10.0.0.2)  1.194 ms  1.058 ms  1.033 ms
 3  10.1.12.2 (10.1.12.2)  1.073 ms  0.949 ms  0.932 ms
```

Example: IKE Dynamic SA Between an AS PIC and an ES PIC Configuration

Figure 7: AS PIC to ES PIC IKE Dynamic SA Topology Diagram

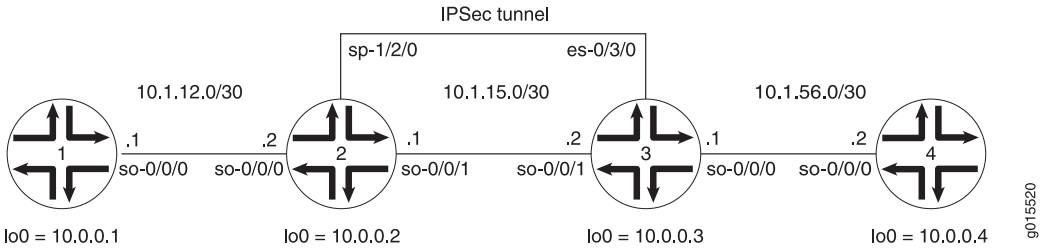


Figure 7 on page 79 shows a hybrid configuration that allows you to create an IPSec tunnel between the AS PIC and the ES PIC. Router 2 contains an AS PIC at **sp-1/2/0** and Router 3 has an ES PIC at **es-0/3/0**. To establish an IPSec tunnel using an IKE dynamic SA, the key is to learn the default IKE SA and IPSec SA settings built into the AS PIC and configure them explicitly on the ES PIC. Routers 1 and 4 again provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
        area 0.0.0.0 {
            interface so-0/0/0.0;
            interface lo0.0;
        }
    }
}
```

On Router 2, enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA in a rule called **rule-ike** at the **[edit ipsec-vpn rule]** hierarchy level. Reference this rule in a service set called **service-set-dynamic-BiEspsha3des** at the **[edit services service-set]** hierarchy level.

Using default values in the AS PIC, you do not need to specify an IPSec proposal, IPSec policy, or IKE proposal. However, you do need to configure a preshared key in an IKE policy with the **pre-shared-key** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level. (For more information about default IKE and IPSec policies and proposals on the AS PIC, see "IKE and IPSec Proposal and Policy Default Values for the AS and MultiServices PICs" on page 20.)

To direct traffic into the AS PIC and the IPSec tunnel, include match conditions in the **rule-ike** IPSec VPN rule to match inbound traffic from Router 1 that is destined for Router 4. Because the rule is already referenced by the service set, apply the service set to the **so-0/0/1** interface. To count the amount of traffic that enters the IPSec tunnel, configure a firewall filter called **ipsec-tunnel** and apply it to the **sp-1/2/0** interface.

Router 2

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R1 so-0/0/0";
        unit 0 {
            family inet {
```

```
            address 10.1.12.1/30;
          }
        }
      }
      so-0/0/1 {
        description "To R3 so-0/0/1";
        unit 0 {
          family inet {
            service { # Apply the service set here.
              input {
                service-set service-set-dynamic-BiEspsha3des;
              }
              output {
                service-set service-set-dynamic-BiEspsha3des;
              }
            }
            address 10.1.15.1/30;
          }
        }
      }
      sp-1/2/0 {
        services-options {
          syslog {
            host local {
              services info;
            }
          }
        }
        unit 0 {
          family inet {
            filter {
              input ipsec-tunnel; # Apply the firewall filter with the counter here.
            }
          }
        }
      }
      lo0 {
        unit 0 {
          family inet {
            address 10.0.0.2/32;
          }
        }
      }
    }
    routing-options {
      router-id 10.0.0.2;
    }
    protocols {
      ospf {
        area 0.0.0.0 {
          interface so-0/0/0.0;
          interface so-0/0/1.0;
          interface lo0.0;
        }
      }
    }
```

```
firewall {
    filter ipsec-tunnel { # Configure a firewall filter to count IPSec traffic here.
        term 1 {
            then {
                count ipsec-tunnel;
                accept;
            }
        }
    }
}
services {
    service-set service-set-dynamic-BiEspsha3des { # Define your service set here.
        interface-service {
            service-interface sp-1/2/0; # Specify an interface to process IPSec.
        }
        ipsec-vpn-options {
            local-gateway 10.1.15.1; # Specify the local IP address of the IPSec tunnel.
        }
        ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
    }
    ipsec-vpn {
        rule rule-ike { # Define your IPSec VPN rule here.
            term term-ike {
                from {
                    source-address {
                        10.1.12.0/24;
                    }
                    destination-address {
                        10.1.56.0/24;
                    }
                }
                then {
                    remote-gateway 10.1.15.2; # The remote IP address of the IPSec tunnel.
                    dynamic { # This creates a dynamic SA.
                        ike-policy ike-policy-preshared; # Reference your IKE proposal here.
                    }
                }
            }
            match-direction output; # Specify in which direction the rule should match.
        }
        ike {
            policy ike-policy-preshared { # Define your IKE policy specifications here.
                pre-shared-key ascii-text "$9$KtKWX-YgJHqfVwqfTzCAvWL";
                ## The unencrypted preshared key for this example is juniper.
            }
        }
    }
}
```

On Router 3, enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA called **sa-dynamic** at the **[edit security ipsec security-association]** hierarchy level. To allow the ES PIC to communicate with the IKE dynamic SA established on Router 2, you must explicitly configure the same policies and proposals on the ES PIC that are available by default on the AS PIC. (For more information

about default IKE and IPSec policies and proposals on the AS PIC, see "IKE and IPSec Proposal and Policy Default Values for the AS and MultiServices PICs" on page 20.)

For your IKE policy and proposal, use preshared keys for the authentication method, SHA-1 for the authentication algorithm, 3DES-CBC for encryption, group 2 for the Diffie-Hellman group, main mode, 3600 seconds for the lifetime, and a preshared key of **juniper** for the initial IKE negotiation. For your IPSec policy and proposal, use ESP for the protocol, HMAC-SHA1-96 for authentication, 3DES-CBC for encryption, 28800 seconds for the lifetime, and group 2 for the PFS group.

To direct traffic into the ES PIC and the IPSec tunnel, create two firewall filters. The **es-traffic** filter matches inbound traffic from Router 4 destined for Router 1, whereas the **es-return** filter matches the return path from Router 1 to Router 4. Apply the **es-traffic** filter to the **so-0/0/0** interface; then apply both the **es-return** filter and the **sa-dynamic SA** to the **es-0/3/0** interface.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R4 so-0/0/0";
        unit 0 {
            family inet {
                filter {
                    input es-traffic; # Apply a filter that sends traffic to the IPSec tunnel here.
                }
                address 10.1.56.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R2 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    es-0/3/0 {
        unit 0 {
            tunnel { # Specify the IPSec tunnel endpoints here.
                source 10.1.15.2;
                destination 10.1.15.1;
            }
            family inet {
                ipsec-sa sa-dynamic; # Apply the dynamic SA here.
                filter {
                    input es-return; # Apply the filter that matches return IPSec traffic here.
                }
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
```

```
            address 10.0.0.3/32;
          }
        }
      }
    }
    routing-options {
      router-id 10.0.0.3;
    }
    protocols {
      ospf {
        area 0.0.0.0 {
          interface so-0/0/0.0;
          interface so-0/0/1.0;
          interface lo0.0;
        }
      }
    }
    security {
      ipsec {
        proposal es-ipsec-proposal { # Define your IPSec proposal specifications here.
          protocol esp;
          authentication-algorithm hmac-sha1-96;
          encryption-algorithm 3des-cbc;
          lifetime-seconds 28800;
        }
        policy es-ipsec-policy { # Define your IPSec policy specifications here.
          perfect-forward-secrecy {
            keys group2;
          }
          proposals es-ipsec-proposal; # Reference the IPSec proposal here.
        }
        security-association sa-dynamic { # Define your dynamic SA here.
          mode tunnel;
          dynamic {
            ipsec-policy es-ipsec-policy; # Reference the IPSec policy here.
          }
        }
      }
      ike {
        proposal es-ike-proposal { # Define your IKE proposal specifications here.
          authentication-method pre-shared-keys;
          dh-group group2;
          authentication-algorithm sha1;
          encryption-algorithm 3des-cbc;
          lifetime-seconds 3600;
        }
        policy 10.1.15.1 { # Define your IKE policy specifications here.
          mode main;
          proposals es-ike-proposal; # Reference the IKE proposal here.
          pre-shared-key ascii-text "$9$TF6ABIcvWxp0WxNdg4QFn";
          ## The unencrypted preshared key for this example is juniper.
        }
      }
    }
    firewall {
      filter es-traffic { # Define a filter that sends traffic to the IPSec tunnel here.
```

```
    term to-es {
      from {
        source-address {
          10.1.56.0/24;
        }
        destination-address {
          10.1.12.0/24;
        }
      }
      then {
        count ipsec-tunnel;
        ipsec-sa sa-dynamic;
      }
    }
    term other {
      then accept;
    }
  }
  filter es-return { # Define a filter that matches return IPSec traffic here.
    term return {
      from {
        source-address {
          10.1.12.0/24;
        }
        destination-address {
          10.1.56.0/24;
        }
      }
      then accept;
    }
  }
}
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R3 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.56.2/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.4/32;
      }
    }
  }
}
routing-options {
  router-id 10.0.0.4;
}
```

```
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface lo0.0;
    }
  }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the AS PIC, use the following commands:

- **ping**
- **show services ipsec-vpn ike security-associations (detail)**
- **show services ipsec-vpn ipsec security-associations (detail)**
- **traceroute**

To verify proper operation of an IKE-based dynamic SA on the ES PIC, use the following commands:

- **ping**
- **show ike security-associations (detail)**
- **show ipsec security-associations (detail)**
- **traceroute**

The following sections show the output of these commands used with the configuration example:

- Router 1 on page 86
- Router 2 on page 87
- Router 3 on page 88
- Router 4 on page 89

Router 1

On Router 1, issue a **ping** command to the **so-0/0/0** interface of Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=253 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=253 time=1.020 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=253 time=0.998 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=253 time=1.037 ms
^C
--- 10.1.56.2 ping statistics ---
```

```
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.998/1.057/1.172/0.068 ms
```

You can also issue the **traceroute** command to verify that traffic to **10.1.56.2** travels over the IPSec tunnel between Router 2 and Router 3. Notice that the traced path does not reference **10.1.15.2**—the physical interface on Router 3. Instead, traffic arriving at Router 2 is immediately filtered into the IPSec tunnel and the path is listed as unknown with the *** notation. This indicates that the IPSec tunnel is operating correctly.

```
user@R1> traceroute 10.1.56.2
traceroute to 10.1.56.2 (10.1.56.2), 30 hops max, 40 byte packets
 1  * * *
 2  10.1.56.2 (10.1.56.2)  1.045 ms  0.915 ms  0.850 ms
```

Router 2

One way to verify that matched traffic is being diverted to the bidirectional IPSec tunnel is to view the firewall filter counter. Before any traffic flows, the **ipsec-tunnel** firewall filter counter looks like this:

```
user@R2> show firewall filter ipsec-tunnel
Filter: ipsec-tunnel
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                            0                    0
```

After you issue the **ping** command from Router 1 (four packets) to **10.1.56.2**, the **ipsec-tunnel** firewall filter counter looks like this:

```
user@R2> show firewall filter ipsec-tunnel
Filter: ipsec-tunnel
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          336                    4
```

After you issue the **ping** command from both Router 1 to **10.1.56.2** (four packets) and from Router 4 to **10.1.12.2** (six packets), the **ipsec-tunnel** firewall filter counter looks like this:

```
user@R2> show firewall filter ipsec-tunnel
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          840                   10
```

To verify that the IKE SA negotiation is successful, issue the **show services ipsec-vpn ike security-associations detail** command. Notice that the SA contains the default IKE settings inherent in the AS PIC, such as SHA-1 for the authentication algorithm and 3DES-CBC for the encryption algorithm.

```
user@R2> show services ipsec-vpn ike security-associations detail
IKE peer 10.1.15.2
  Role: Responder, State: Matured
  Initiator cookie: c8e1e4c0da000040, Responder cookie: 4fbaa5184e000044
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.1:500, Remote: 10.1.15.2:500
  Lifetime: Expires in 3535 seconds
  Algorithms:
   Authentication        : sha1
```

```
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                  840
   Output bytes  :                  756
   Input  packets:                    5
   Output packets:                    4
  Flags: Caller notification sent
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 0
```

To verify that the IPSec security association is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. Notice that the SA contains the default settings inherent in the AS PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des
  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
    Direction: inbound, SPI: 407204513, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 24546 seconds
    Hard lifetime: Expires in 24636 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 2957235894, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 24546 seconds
    Hard lifetime: Expires in 24636 seconds
    Anti-replay service: Disabled
```

Router 3

View the firewall filter counter to continue verifying that matched traffic is being diverted to the bidirectional IPSec tunnel. After you issue the **ping** command from Router 1 (four packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          336                    4
```

After you issue the **ping** command from both Router 1 (four packets) and Router 4 (six packets), the **es-traffic** firewall filter counter looks like this:

```
user@R3> show firewall filter es-traffic
Filter: es-traffic
Counters:
Name                                                Bytes              Packets
ipsec-tunnel                                          840                   10
```

To verify the success of the IKE security association on the ES PIC, issue the **show ike security-associations detail** command. Notice that the IKE SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ike security-associations detail
IKE peer 10.1.15.1
  Role: Initiator, State: Matured
  Initiator cookie: c8e1e4c0da000040, Responder cookie: 4fbaa5184e000044
  Exchange type: Main, Authentication method: Pre-shared-keys
  Local: 10.1.15.2:500, Remote: 10.1.15.1:500
  Lifetime: Expires in 3441 seconds
  Algorithms:
   Authentication        : sha1
   Encryption            : 3des-cbc
   Pseudo random function: hmac-sha1
  Traffic statistics:
   Input  bytes  :                  756
   Output bytes  :                  840
   Input  packets:                    4
   Output packets:                    5
  Flags: Caller notification sent
  IPSec security associations: 1 created, 0 deleted
  Phase 2 negotiations in progress: 0
```

To verify that the IPSec security association is active, issue the **show ipsec security-associations detail** command. Notice that the IPSec SA on Router 3 contains the same settings you specified on Router 2.

```
user@R3> show ipsec security-associations detail
Security association: sa-dynamic, Interface family: Up
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  Local identity: ipv4_subnet(any:0,[0..7]=10.1.56.0/24)
  Remote identity: ipv4_subnet(any:0,[0..7]=10.1.12.0/24)
    Direction: inbound, SPI: 2957235894, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 28555 seconds
    Hard lifetime: Expires in 28690 seconds
    Anti-replay service: Disabled
    Direction: outbound, SPI: 407204513, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 28555 seconds
    Hard lifetime: Expires in 28690 seconds
    Anti-replay service: Disabled
```

Router 4

On Router 4, issue a **ping** command to the **so-0/0/0** interface on Router 1 to send traffic across the IPSec tunnel.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=3 ttl=254 time=1.142 ms
64 bytes from 10.1.12.2: icmp_seq=4 ttl=254 time=1.139 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

Again, the **traceroute** command verifies that traffic to **10.1.12.2** travels over the IPSec tunnel between Router 3 and Router 2. Notice that the second hop does not reference **10.1.15.1**—the physical interface on Router 2. Instead, the second hop is listed as unknown with the *** notation. This indicates that the IPSec tunnel is operating correctly.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.56.1 (10.1.56.1)  3.561 ms  0.613 ms  0.558 ms
 2  * * *
 3  10.1.12.2 (10.1.12.2)  1.073 ms  0.862 ms  0.818 ms
```

Example: AS PIC IKE Dynamic SA with Digital Certificates Configuration

Figure 8: AS PIC IKE Dynamic SA Topology Diagram

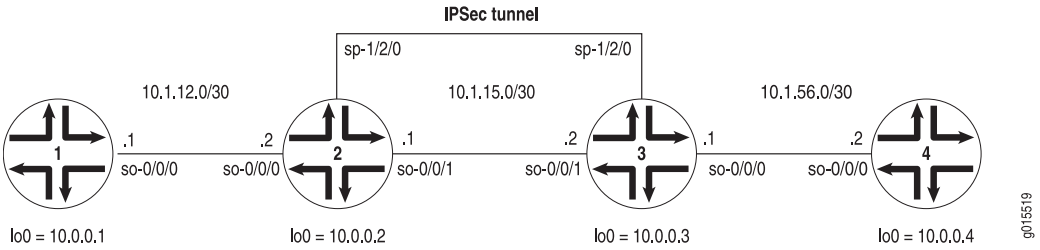


Figure 8 on page 90 shows the same IPSec topology as the AS PIC dynamic SA example on "Example: AS PIC IKE Dynamic SA Configuration" on page 71. However, this configuration requires Routers 2 and 3 to establish an IKE-based IPSec tunnel by using digital certificates in place of preshared keys. Routers 1 and 4 continue to provide basic connectivity and are used to verify that the IPSec tunnel is operational.

On Router 1, provide basic OSPF connectivity to Router 2.

Router 1

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R2 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.12.2/30;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.1/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.1;
}
protocols {
    ospf {
```

```
      area 0.0.0.0 {
        interface so-0/0/0.0;
        interface lo0.0;
      }
    }
  }
```

On Router 2, you must request a CA certificate, create a local certificate, and load these digital certificates into the router before you can reference them in your IPSec configuration. To begin, configure an IPSec profile by specifying the trusted CA and URL of the CA server that handles CA certificate processing:

```
[edit]
security {
  pki {
    ca-profile entrust {
      ca-identity entrust;
      enrollment {
        url http://ca-1.jnpr.net/cgi-bin/pkiclient.exe;
      }
    }
  }
}
```

Certificate revocation list (CRL) verification is enabled by default. You can optionally specify the Lightweight Access Directory (LDAP) server where the CA stores the CRL. The certificate typically includes a certificate distribution point (CDP), which contains information about how to retrieve the CRL for the certificate. The router uses this information to download the CRL automatically. In this example, the LDAP URL is specified, which overrides the location provided in the certificate:

```
[edit]
security pki ca-profile entrust {
  revocation-check {
    crl {
      url ldap://10.157.90.185/o=juniper,c=uscertificateRevocationListbase;
    }
  }
}
```

After you configure the CA profile, you can request a CA certificate from the trusted CA. In this example, the certificate is enrolled online and installed into the router automatically.

```
user@R2> request security pki ca-certificate enroll ca-profile entrust
Received following certificates:
  Certificate: C=us, O=juniper
    Fingerprint: 00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: 46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f
Do you want to load the above CA certificate ? [yes,no] (no) yes
```

NOTE: If you obtain the CA certificate directly from the CA (for example, as an e-mail attachment or website download), you can install it with the **request security pki ca-certificate load** command.

Next, you must generate a private/public key pair before you can create a local certificate.

```
user@R2> request security pki generate-key-pair certificate-id local-entrust2
Generated key pair local-entrust2, key size 1024 bits
```

When the key pair is available, generate a local certificate request and send it to the CA for processing.

```
user@R2> request security pki generate-certificate-request
certificate-id local-entrust2 domain-name router2.juniper.net
filename entrust-req2 subject cn=router2.juniper.net
Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIBoTCCAQoCAQAwGjEYMBYGA1UEAxMPdHAxLmp1bmlwZXIubmV0MIGfMA0GCSqG
SIb3DQEBAQUAA4GNADCBiQKBgQCiUFklQws1Ud+AqN5DDxRs2kVyKEhh9qoVFnz+
Hz4c9vsy3B8ElwTJlkmIt2cB3yifB6zePd+6WYpf57Crwre7YqPkiXM31F6z3YjX
H+1BPNbCxNWYvyrnSyVYDbFj8oOXyqog8ACDfVL2JBWrPNBYy7imq/K9soDBbAs6
5hZqqwIDAQABoEcwRQYJKoZIhvcNAQkOMTgwNjAOBgNVHQ8BAf8EBAMCB4AwJAYD
VR0RAQH/BBowGIIWdHAxLmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQF
AAOBgQBc2rq1v5SOQXH7LCb/FdqAL8ZM6GoaN5d6cGwq4bB6a7UQFgtoH406gQ3G
3iHOZfz4xMIBpJYuGd1dkqgvcDoH3AgTsLkfn7Wi3x5H2qeQVs9bvL4P5nvEZLND
EIMUHwteolZCiZ7OfO9Fer9cXWHSQs1UtXtgPqQJy2xIeImLgw==
-----END CERTIFICATE REQUEST-----
Fingerprint:
0d:90:b8:d2:56:74:fc:84:59:62:b9:78:71:9c:e4:9c:54:ba:16:97 (sha1)
1b:08:d4:f7:90:f1:c4:39:08:c9:de:76:00:86:62:b8 (md5)
```

NOTE: You can request the creation and installation of a local certificate online with the **request security pki local-certificate enroll** command. For more information, see "Generating and Enrolling a Local Digital Certificate" on page 28 or the *Junos System Basics and Services Command Reference*.

The trusted CA digitally signs the local certificate and returns it to you. Copy the certificate file into the router and load the certificate.

```
user@R2> request security pki local-certificate load filename /tmp/router2-cert   certificate-id
local-entrust2
Local certificate local-entrust2 loaded successfully
```

NOTE: The name of the file sent to you by the CA might not match the name of the certificate identifier. However, the **certificate-id** name must always match the name of the key pair you generated for the router.

After the local and CA certificates have been loaded, you can reference them in your IPSec configuration.

Using default values in the AS PIC, you do not need to configure an IPSec proposal or IPSec policy. However, you must configure an IKE proposal that specifies the use of digital certificates, reference the IKE proposal and local certificate in an IKE policy, and apply the CA profile to the service set. To enable an IKE proposal for digital certificates, include the **rsa-signatures** statement at the **[edit services ipsec-vpn ike proposal *proposal-name* authentication-method]** hierarchy level. To reference the local certificate in the IKE policy, include the **local-certificate** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level. To identify the CA or RA in the service set, include the **trusted-ca** statement at the **[edit services service-set *service-set-name* ipsec-vpn-options]** hierarchy level.

NOTE: For more information about default IKE and IPSec policies and proposals on the AS PIC, see "IKE and IPSec Proposal and Policy Default Values for the AS and MultiServices PICs" on page 20.

Optionally, you can configure automatic reenrollment of the certificate with the **auto-re-enrollment** statement at the [**edit security pki**] hierarchy level.

The remaining configuration components of your IKE-based IPSec tunnel are the same as when you use preshared keys. Enable OSPF as the underlying routing protocol to connect to Routers 1 and 3. Configure a bidirectional IKE dynamic SA in a rule called **rule-ike** at the **[edit ipsec-vpn rule]** hierarchy level. Reference this rule in a service set called **service-set-dynamic-BiEspsha3des** at the **[edit services service-set]** hierarchy level.

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 2

```
[edit]
interfaces {
   so-0/0/0 {
      description "To R1 so-0/0/0";
      unit 0 {
         family inet {
            address 10.1.12.1/30;
         }
      }
   }
   so-0/0/1 {
      description "To R3 so-0/0/1";
      unit 0 {
         family inet {
            address 10.1.15.1/30;
         }
      }
   }
   sp-1/2/0 {
      unit 0 {
         family inet;
      }
      unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
         family inet;
         service-domain inside;
```

```
      }
      unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
        family inet;
        service-domain outside;
      }
    }
    lo0 {
      unit 0 {
        family inet {
          address 10.0.0.2/32;
        }
      }
    }
}
routing-options {
  router-id 10.0.0.2;
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
      interface lo0.0;
    }
  }
}
security { # Configure CA profiles here, including the URLs used to reach the CAs.
  pki {
    ca-profile entrust {
      ca-identity entrust;
      enrollment {
        url http://ca-1.jnpr.net/cgi-bin/pkiclient.exe;
      }
      revocation-check {
        crl {
          url ldap://10.157.90.185/o=juniper,c=uscertificateRevocationListbase;
          # Specify the URL of the LDAP server where the CA stores the CRL.
        }
      }
    }
    ca-profile microsoft {
      ca-identity microsoft;
      enrollment {
        url http://192.168.11.78:80/certsrv/mscep/mscep.dll;
      }
    }
    ca-profile verisign {
      ca-identity verisign;
      enrollment {
        url http://pilotonsiteipsec.verisign.com/cgi-bin/pkiclient.exe;
      }
    }
  }
}
services {
  service-set service-set-dynamic-BiEspsha3des { # Define your service set here.
```

```
      next-hop-service { # Required for dynamic routing protocols such as OSPF.
        inside-service-interface sp-1/2/0.1;
        outside-service-interface sp-1/2/0.2;
      }
      ipsec-vpn-options {
        trusted-ca entrust; # Reference the CA profile here.
        local-gateway 10.1.15.1; # Specify the local IP address of the IPSec tunnel.
      }
      ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
    }
    ipsec-vpn {
      rule rule-ike { # Define your IPSec VPN rule here.
        term term-ike {
          then {
            remote-gateway 10.1.15.2; # The remote IP address of the IPSec tunnel.
            dynamic { # This creates a dynamic SA.
              ike-policy ike-digital-certificates; # Reference your IKE policy here.
            }
          }
        }
        match-direction input; # Specify in which direction the rule should match.
      }
      ike {
        proposal ike-proposal {
          authentication-method rsa-signatures; # Uses digital certificates
        }
        policy ike-digital-certificates {
          proposals ike-proposal; # Apply the IKE proposal here.
          local-id fqdn router2.juniper.net; # Provide an identifier for the local router.
          local-certificate local-entrust2; # Reference the local certificate here.
          remote-id fqdn router3.juniper.net; # Provide an ID for the remote router.
        }
      }
      establish-tunnels immediately;
    }
  }
```

On Router 3, you must repeat the digital certificate procedures you performed on Router 2. If the IPSec peers do not have a symmetrical configuration containing all the necessary components, they cannot establish a peering relationship.

You need to request a CA certificate, create a local certificate, load these digital certificates into the router, and reference them in your IPSec configuration. Begin by configuring an IPSec CA profile. Include the **ca-profile** statement at the **[edit security pki]** hierarchy level and specify the trusted CA and URL of the CA server that handles CA certificate processing. Include the CRL statements found on Router 2 to complete your CA profile on Router 3.

After you configure the CA profile, request a CA certificate from the trusted CA. In this example, the certificate is enrolled online and installed into the router automatically.

```
user@R3> request security pki ca-certificate enroll ca-profile entrust
Received following certificates:
  Certificate: C=us, O=juniper
    Fingerprint: 00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10
  Certificate: C=us, O=juniper, CN=First Officer
```

```
    Fingerprint: bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17
  Certificate: C=us, O=juniper, CN=First Officer
    Fingerprint: 46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f
Do you want to load the above CA certificate ? [yes,no] (no) yes
```

NOTE: If you obtain the CA certificate directly from the CA (for example, as an e-mail attachment or website download), you can install it with the **request security pki ca-certificate load** command.

Next, generate a private/public key pair.

```
user@R3> request security pki generate-key-pair certificate-id local-entrust3
Generated key pair local-entrust3, key size 1024 bits
```

When the key pair is available, you can generate a local certificate request and send it to the CA for processing.

```
user@R3> request security pki generate-certificate-request
certificate-id local-entrust3 domain-name router3.juniper.net
filename entrust-req3 subject cn=router3.juniper.net
Generated certificate request
-----BEGIN CERTIFICATE REQUEST-----
MIIB8jCCAVsCAQAwZTEYMBYGA1UEAxMPdHA1Lmp1bmlwZXIubmV0MRQwEgYDVQQL
EwtFbmdpbmVlcmluZzEQMA4GA1UEChMHSnVuaXBlcjETMBEGA1UECBMKQ2FsaWZv
cm5pYTEMMAoGA1UEBhMDVVNBMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCg
Wjo5Ow8jrnphsOsRFvqQMwC6PlYa65thrJ8nHZ2qgYgRbSrO8hdODhvU6/5VuD2/
zBtgV5ZSAOlyV6DXqlbVj/2XirQAJMRCr1eYu6DhYRBMNq/UaQv4Z8Sse1EJv+uR
HTNbD7xlwpw2zwzltRuGFtFr/FrGBOhF7IE+Xm5e2wIDAQABoE0wSwYJKoZIhvcN
AQkOMT4wPDAOBgNVHQ8BAf8EBAMCB4AwKgYDVR0RAQH/BCAwHocEwKhGk4IWdHA1
LmVuZ2xhYi5qdW5pcGVyLm5ldDANBgkqhkiG9w0BAQQFAAOBgQBbiJ+ZCeQ59/eY
4Rd6awIpJFTzOsvRZLxxjFWogusVTmaD2dsqFBqftS1eJBdeiueRcYMF9vOnOGKm
FNfouegwei5+vzdNmNo55eIb3rs4pP62qOW5CUgmbHrjtp3lyJsvuOxTTcPNY8zw
b6GyM2Hdkk3Vh2ReX11tQUSqYujTjw==
-----END CERTIFICATE REQUEST-----
Fingerprint:
7c:e8:f9:45:93:8d:a3:92:7f:18:29:02:f1:c8:e2:85:3d:ad:df:1f (sha1)
00:4e:df:a0:6b:ad:8c:50:da:7c:a1:cf:5d:37:b0:ea (md5)
```

The trusted CA digitally signs the local certificate and returns it to you. Copy the certificate file into the router and load the certificate.

```
user@R3> request security pki local-certificate load filename /tmp/router3-cert certificate-id
local-entrust3
Local certificate local-entrust3 loaded successfully
```

After the local and CA certificates have been loaded, you can reference them in your IPSec configuration. Using default values in the AS PIC, you do not need to configure an IPSec proposal or IPSec policy. However, you must configure an IKE proposal that uses digital certificates, reference the IKE proposal and local certificate in an IKE policy, and apply the CA profile to the service set. To enable the IKE proposal for digital certificates, include the **rsa-signatures** statement at the **[edit services ipsec-vpn ike proposal *proposal-name* authentication-method]** hierarchy level. To reference the local certificate in the IKE policy, include the **local-certificate** statement at the **[edit services ipsec-vpn ike policy *policy-name*]** hierarchy level. To identify the CA or RA in the service set, include the **trusted-ca** statement at the **[edit services service-set *service-set-name* ipsec-vpn-options]** hierarchy level.

The remaining configuration components of your IKE-based IPSec tunnel are the same as when you use preshared keys. Enable OSPF as the underlying routing protocol to connect to Routers 2 and 4. Configure a bidirectional IKE dynamic SA in a rule called **rule-ike** at the **[edit ipsec-vpn rule]** hierarchy level. Reference this rule in a service set called **service-set-dynamic-BiEspsha3des** at the **[edit services service-set]** hierarchy level.

To direct traffic into the AS PIC and the IPSec tunnel, configure a next-hop style service set and add the adaptive services logical interface used as the IPSec inside interface into the OSPF configuration.

Router 3

```
[edit]
interfaces {
    so-0/0/0 {
        description "To R4 so-0/0/0";
        unit 0 {
            family inet {
                address 10.1.56.1/30;
            }
        }
    }
    so-0/0/1 {
        description "To R2 so-0/0/1";
        unit 0 {
            family inet {
                address 10.1.15.2/30;
            }
        }
    }
    sp-1/2/0 {
        unit 0 {
            family inet;
        }
        unit 1 { # sp-1/2/0.1 is the IPSec inside interface.
            family inet;
            service-domain inside;
        }
        unit 2 { # sp-1/2/0.2 is the IPSec outside interface.
            family inet;
            service-domain outside;
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 10.0.0.3/32;
            }
        }
    }
}
routing-options {
    router-id 10.0.0.3;
}
protocols {
    ospf {
        area 0.0.0.0 {
```

```
        interface so-0/0/0.0;
        interface sp-1/2/0.1; # This sends OSPF traffic over the IPSec tunnel.
        interface lo0.0;
      }
    }
}
security { # Configure CA profiles here, including the URLs used to reach the CAs.
    pki {
      ca-profile entrust {
        ca-identity entrust;
        enrollment {
          url http://ca-1.jnpr.net/cgi-bin/pkiclient.exe;
        }
        revocation-check {
          crl {
            url ldap://10.157.90.185/o=juniper,c=uscertificateRevocationListbase;
            # Specify the URL of the LDAP server where the CA stores the CRL.
          }
        }
      }
      ca-profile microsoft {
        ca-identity microsoft;
        enrollment {
          url http://192.168.11.78:80/certsrv/mscep/mscep.dll;
        }
      }
      ca-profile verisign {
        ca-identity verisign;
        enrollment {
          url http://pilotonsiteipsec.verisign.com/cgi-bin/pkiclient.exe;
        }
      }
    }
}
services {
    service-set service-set-dynamic-BiEspsha3des { # Define your service set here.
      next-hop-service { # Required for dynamic routing protocols such as OSPF.
        inside-service-interface sp-1/2/0.1;
        outside-service-interface sp-1/2/0.2;
      }
      ipsec-vpn-options {
        trusted-ca entrust; # Reference the CA profile here.
        local-gateway 10.1.15.2; # Specify the local IP address of the IPSec tunnel.
      }
      ipsec-vpn-rules rule-ike; # Reference your IPSec VPN rule here.
    }
    ipsec-vpn {
      rule rule-ike { # Define your IPSec VPN rule here.
        term term-ike {
          then {
            remote-gateway 10.1.15.1; # The remote IP address of the IPSec tunnel.
            dynamic { # This creates a dynamic SA.
              ike-policy ike-digital-certificates; # Reference your IKE policy here.
            }
          }
        }
```

```
        match-direction input; # Specify in which direction the rule should match.
      }
      ike {
        proposal ike-proposal {
          authentication-method rsa-signatures; # Uses digital certificates
        }
        policy ike-digital-certificates {
          proposals ike-proposal; # Apply the IKE proposal here.
          local-id fqdn router3.juniper.net; # Provide an identifier for the local router.
          local-certificate local-entrust3; # Reference the local certificate here.
          remote-id fqdn router2.juniper.net; # Provide an ID for the remote router.
        }
      }
      establish-tunnels immediately;
    }
  }
```

On Router 4, provide basic OSPF connectivity to Router 3.

Router 4

```
[edit]
interfaces {
  so-0/0/0 {
    description "To R3 so-0/0/0";
    unit 0 {
      family inet {
        address 10.1.56.2/30;
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 10.0.0.4/32;
      }
    }
  }
}
routing-options {
  router-id 10.0.0.4;
}
protocols {
  ospf {
    area 0.0.0.0 {
      interface so-0/0/0.0;
      interface lo0.0;
    }
  }
}
```

Verifying Your Work

To verify proper operation of an IKE-based dynamic SA on the AS PIC, use the following commands:

- **ping**
- **show services ipsec-vpn certificates (detail)**
- **show services ipsec-vpn ike security-associations (detail)**
- **show services ipsec-vpn ipsec security-associations (detail)**
- **show services ipsec-vpn ipsec statistics**
- **traceroute**

To verify and manage digital certificates in your router, use the following commands:

- **show security pki ca-certificate (detail)**
- **show security pki certificate-request (detail)**
- **show security pki local-certificate (detail)**

The following sections show the output of these commands used with the configuration example:

- Router 1 on page 100
- Router 2 on page 101
- Router 3 on page 104
- Router 4 on page 107

Router 1

On Router 1, issue a **ping** command to the **so-0/0/0** interface on Router 4 to send traffic across the IPSec tunnel.

```
user@R1> ping 10.1.56.2
PING 10.1.56.2 (10.1.56.2): 56 data bytes
64 bytes from 10.1.56.2: icmp_seq=0 ttl=254 time=1.351 ms
64 bytes from 10.1.56.2: icmp_seq=1 ttl=254 time=1.187 ms
64 bytes from 10.1.56.2: icmp_seq=2 ttl=254 time=1.172 ms
64 bytes from 10.1.56.2: icmp_seq=3 ttl=254 time=1.154 ms
64 bytes from 10.1.56.2: icmp_seq=4 ttl=254 time=1.156 ms
^C
--- 10.1.56.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.154/1.204/1.351/0.074 ms
```

If you ping the loopback address of Router 4, the operation succeeds because the address is part of the OSPF network configured on Router 4.

```
user@R1> ping 10.0.0.4
PING 10.0.0.4 (10.0.0.4): 56 data bytes
64 bytes from 10.0.0.4: icmp_seq=0 ttl=62 time=1.318 ms
64 bytes from 10.0.0.4: icmp_seq=1 ttl=62 time=1.084 ms
64 bytes from 10.0.0.4: icmp_seq=2 ttl=62 time=3.260 ms
```

```
^C
--- 10.0.0.4 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.084/1.887/3.260/0.975 ms
```

Router 2

To verify that matched traffic is being diverted to the bidirectional IPSec tunnel, view the IPSec statistics:

```
user@R2> show services ipsec-vpn ipsec statistics

PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des

ESP Statistics:
  Encrypted bytes:           162056
  Decrypted bytes:           161896
  Encrypted packets:           2215
  Decrypted packets:           2216
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

To verify that the IKE SA negotiation is successful, issue the **show services ipsec-vpn ike security-associations** command:

```
user@R2> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.2       Matured       d82610c59114fd37  ec4391f76783ef28  Main
```

To verify that the IPSec security association is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. Notice that the SA contains the default settings inherent in the AS PIC, such as ESP for the protocol and HMAC-SHA1-96 for the authentication algorithm.

```
user@R2> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des

  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.1, Remote gateway: 10.1.15.2
  IPSec inside interface: sp-1/2/0.1
  Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)

    Direction: inbound, SPI: 857451461, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 9052 seconds
    Hard lifetime: Expires in 9187 seconds
    Anti-replay service: Enabled, Replay window size: 64

    Direction: outbound, SPI: 1272330309, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
```

```
    Soft lifetime: Expires in 9052 seconds
    Hard lifetime: Expires in 9187 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To display the digital certificates that are used to establish the IPSec tunnel, issue the **show services ipsec-vpn certificates** command:

```
user@R2> show services ipsec-vpn certificates
Service set: service-set-dynamic-BiEspsha3des, Total entries: 3
  Certificate cache entry: 3
    Flags: Non-root Trusted
    Issued to: router3.juniper.net, Issued by: juniper
    Alternate subject: router3.juniper.net
    Validity:
      Not before: 2005 Nov 21st, 23:33:58 GMT
      Not after: 2008 Nov 22nd, 00:03:58 GMT

  Certificate cache entry: 2
    Flags: Non-root Trusted
    Issued to: router2.juniper.net, Issued by: juniper
    Alternate subject: router2.juniper.net
    Validity:
      Not before: 2005 Nov 21st, 23:28:22 GMT
      Not after: 2008 Nov 21st, 23:58:22 GMT

  Certificate cache entry: 1
    Flags: Root Trusted
    Issued to: juniper, Issued by: juniper
    Validity:
      Not before: 2005 Oct 18th, 23:54:22 GMT
      Not after: 2025 Oct 19th, 00:24:22 GMT
```

To display the CA certificate, issue the **show security pki ca-certificate detail** command. Notice that there are three separate certificates: one for certificate signing, one for key encipherment, and one for the CA's digital signature.

```
user@R2> show security pki ca-certificate detail
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 9235
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us
  Validity:
    Not before: 2005 Oct 18th, 23:54:22 GMT
    Not after: 2025 Oct 19th, 00:24:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    cb:9e:2d:c0:70:f8:ea:3c:f2:b5:f0:02:48:87:dc:68:99:a3:57:4f
    0e:b9:98:0b:95:47:0d:1f:97:7c:53:17:dd:1a:f8:da:e5:08:d1:1c
    78:68:1f:2f:72:9f:a2:cf:81:e3:ce:c5:56:89:ce:f0:97:93:fa:36
    19:3e:18:7d:8c:9d:21:fe:1f:c3:87:8d:b3:5d:f3:03:66:9d:16:a7
    bf:18:3f:f0:7a:80:f0:62:50:43:83:4f:0e:d7:c6:42:48:c0:8a:b2
    c7:46:30:38:df:9b:dc:bc:b5:08:7a:f3:cd:64:db:2b:71:67:fe:d8
    04:47:08:07:de:17:23:13
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10 (sha1)
    71:6f:6a:76:17:9b:d6:2a:e7:5a:72:97:82:6d:26:86 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
```

```
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: CRL signing, Certificate signing

Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925c
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    c0:a4:21:32:95:0a:cd:ec:12:03:d1:a2:89:71:8e:ce:4e:a6:f9:2f
    1a:9a:13:8c:f6:a0:3d:c9:bd:9d:c2:a0:41:77:99:1b:1e:ed:5b:80
    34:46:f8:5b:28:34:38:2e:91:7d:4e:ad:14:86:78:67:e7:02:1d:2e
    19:11:b7:fa:0d:ba:64:20:e1:28:4e:3e:bb:6e:64:dc:cd:b1:b4:7a
    ca:8f:47:dd:40:69:c2:35:95:ce:b8:85:56:d7:0f:2d:04:4d:5d:d8
    42:e1:4f:6b:bf:38:c0:45:1e:9e:f0:b4:7f:74:6f:e9:70:fd:4a:78
    da:eb:10:27:bd:46:34:33
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17 (sha1)
    23:79:40:c9:6d:a6:f0:ca:e0:13:30:d4:29:6f:86:79 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Key encipherment
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925b
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    ea:75:c4:f3:58:08:ea:65:5c:7e:b3:de:63:0a:cf:cf:ec:9a:82:e2
    d7:e8:b9:2f:bd:4b:cd:86:2f:f1:dd:d8:a2:95:af:ab:51:a5:49:4e
    00:10:c6:25:ff:b5:49:6a:99:64:74:69:e5:8c:23:5b:b4:70:62:8e
    e4:f9:a2:28:d4:54:e2:0b:1f:50:a2:92:cf:6c:8f:ae:10:d4:69:3c
    90:e2:1f:04:ea:ac:05:9b:3a:93:74:d0:59:24:e9:d2:9d:c2:ef:22
    b9:32:c7:2c:29:4f:91:cb:5a:26:fe:1d:c0:36:dc:f4:9c:8b:f5:26
    af:44:bf:53:aa:d4:5f:67
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f (sha1)
    ee:cc:c7:f4:5d:ac:65:33:0a:55:db:59:72:2c:dd:16 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Digital signature
```

To display the local certificate request, issue the **show security pki certificate-request** command:

```
user@R2> show security pki certificate-request
```

```
Certificate identifier: local-entrust2
  Issued to: router2.juniper.net
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

To display the local certificate, issue the **show security pki local-certificate** command:

```
user@R2> show security pki local-certificate
Certificate identifier: local-entrust2
  Issued to: router2.juniper.net, Issued by: juniper
  Validity:
    Not before: 2005 Nov 21st, 23:28:22 GMT
    Not after: 2008 Nov 21st, 23:58:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

Router 3

To verify that matched traffic is being diverted to the bidirectional IPSec tunnel, view the IPSec statistics:

```
user@R3> show services ipsec-vpn ipsec statistics

PIC: sp-1/2/0, Service set: service-set-dynamic-BiEspsha3des

ESP Statistics:
  Encrypted bytes:           161896
  Decrypted bytes:           162056
  Encrypted packets:           2216
  Decrypted packets:           2215
AH Statistics:
  Input bytes:                    0
  Output bytes:                   0
  Input packets:                  0
  Output packets:                 0
Errors:
  AH authentication failures: 0, Replay errors: 0
  ESP authentication failures: 0, ESP decryption failures: 0
  Bad headers: 0, Bad trailers: 0
```

To verify that the IKE SA negotiation is successful, issue the **show services ipsec-vpn ike security-associations** command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ike security-associations
Remote Address  State         Initiator cookie  Responder cookie  Exchange type
10.1.15.1       Matured       d82610c59114fd37  ec4391f76783ef28  Main
```

To verify that the IPSec SA is active, issue the **show services ipsec-vpn ipsec security-associations detail** command. To be successful, the SA on Router 3 must contain the same settings you specified on Router 2.

```
user@R3> show services ipsec-vpn ipsec security-associations detail
Service set: service-set-dynamic-BiEspsha3des

  Rule: rule-ike, Term: term-ike, Tunnel index: 1
  Local gateway: 10.1.15.2, Remote gateway: 10.1.15.1
  IPSec inside interface: sp-1/2/0.1
  Local identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
  Remote identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
```

```
    Direction: inbound, SPI: 1272330309, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 7219 seconds
    Hard lifetime: Expires in 7309 seconds
    Anti-replay service: Enabled, Replay window size: 64

    Direction: outbound, SPI: 857451461, AUX-SPI: 0
    Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 7219 seconds
    Hard lifetime: Expires in 7309 seconds
    Anti-replay service: Enabled, Replay window size: 64
```

To display the digital certificates that are used to establish the IPSec tunnel, issue the **show services ipsec-vpn certificates** command:

```
user@R3> show services ipsec-vpn certificates
Service set: service-set-dynamic-BiEspsha3des, Total entries: 3
  Certificate cache entry: 3
    Flags: Non-root Trusted
    Issued to: router3.juniper.net, Issued by: juniper
    Alternate subject: router3.juniper.net
    Validity:
      Not before: 2005 Nov 21st, 23:33:58 GMT
      Not after: 2008 Nov 22nd, 00:03:58 GMT

  Certificate cache entry: 2
    Flags: Non-root Trusted
    Issued to: router2.juniper.net, Issued by: juniper
    Alternate subject: router2.juniper.net
    Validity:
      Not before: 2005 Nov 21st, 23:28:22 GMT
      Not after: 2008 Nov 21st, 23:58:22 GMT

  Certificate cache entry: 1
    Flags: Root Trusted
    Issued to: juniper, Issued by: juniper
    Validity:
      Not before: 2005 Oct 18th, 23:54:22 GMT
      Not after: 2025 Oct 19th, 00:24:22 GMT
```

To display the CA certificate, issue the **show security pki ca-certificate detail** command. Notice that there are three separate certificates: one for certificate signing, one for key encipherment, and one for the CA's digital signature.

```
user@R3> show security pki ca-certificate detail
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 9235
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us
  Validity:
    Not before: 2005 Oct 18th, 23:54:22 GMT
    Not after: 2025 Oct 19th, 00:24:22 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    cb:9e:2d:c0:70:f8:ea:3c:f2:b5:f0:02:48:87:dc:68:99:a3:57:4f
    0e:b9:98:0b:95:47:0d:1f:97:7c:53:17:dd:1a:f8:da:e5:08:d1:1c
    78:68:1f:2f:72:9f:a2:cf:81:e3:ce:c5:56:89:ce:f0:97:93:fa:36
```

```
    19:3e:18:7d:8c:9d:21:fe:1f:c3:87:8d:b3:5d:f3:03:66:9d:16:a7
    bf:18:3f:f0:7a:80:f0:62:50:43:83:4f:0e:d7:c6:42:48:c0:8a:b2
    c7:46:30:38:df:9b:dc:bc:b5:08:7a:f3:cd:64:db:2b:71:67:fe:d8
    04:47:08:07:de:17:23:13
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    00:8e:6f:58:dd:68:bf:25:0a:e3:f9:17:70:d6:61:f3:53:a7:79:10 (sha1)
    71:6f:6a:76:17:9b:d6:2a:e7:5a:72:97:82:6d:26:86 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: CRL signing, Certificate signing
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925c
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    c0:a4:21:32:95:0a:cd:ec:12:03:d1:a2:89:71:8e:ce:4e:a6:f9:2f
    1a:9a:13:8c:f6:a0:3d:c9:bd:9d:c2:a0:41:77:99:1b:1e:ed:5b:80
    34:46:f8:5b:28:34:38:2e:91:7d:4e:ad:14:86:78:67:e7:02:1d:2e
    19:11:b7:fa:0d:ba:64:20:e1:28:4e:3e:bb:6e:64:dc:cd:b1:b4:7a
    ca:8f:47:dd:40:69:c2:35:95:ce:b8:85:56:d7:0f:2d:04:4d:5d:d8
    42:e1:4f:6b:bf:38:c0:45:1e:9e:f0:b4:7f:74:6f:e9:70:fd:4a:78
    da:eb:10:27:bd:46:34:33
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    bc:78:87:9b:a7:91:13:20:71:db:ac:b5:56:71:42:ad:1a:b6:46:17 (sha1)
    23:79:40:c9:6d:a6:f0:ca:e0:13:30:d4:29:6f:86:79 (md5)
  Distribution CRL:
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Key encipherment
Certificate identifier: entrust
  Certificate version: 3
  Serial number: 4355 925b
  Issuer:
    Organization: juniper, Country: us
  Subject:
    Organization: juniper, Country: us, Common name: First Officer
  Validity:
    Not before: 2005 Oct 18th, 23:55:59 GMT
    Not after: 2008 Oct 19th, 00:25:59 GMT
  Public key algorithm: rsaEncryption(1024 bits)
    ea:75:c4:f3:58:08:ea:65:5c:7e:b3:de:63:0a:cf:cf:ec:9a:82:e2
    d7:e8:b9:2f:bd:4b:cd:86:2f:f1:dd:d8:a2:95:af:ab:51:a5:49:4e
    00:10:c6:25:ff:b5:49:6a:99:64:74:69:e5:8c:23:5b:b4:70:62:8e
    e4:f9:a2:28:d4:54:e2:0b:1f:50:a2:92:cf:6c:8f:ae:10:d4:69:3c
    90:e2:1f:04:ea:ac:05:9b:3a:93:74:d0:59:24:e9:d2:9d:c2:ef:22
    b9:32:c7:2c:29:4f:91:cb:5a:26:fe:1d:c0:36:dc:f4:9c:8b:f5:26
    af:44:bf:53:aa:d4:5f:67
  Signature algorithm: sha1WithRSAEncryption
  Fingerprint:
    46:71:15:34:f0:a6:41:76:65:81:33:4f:68:47:c4:df:78:b8:e3:3f (sha1)
    ee:cc:c7:f4:5d:ac:65:33:0a:55:db:59:72:2c:dd:16 (md5)
  Distribution CRL:
```

```
    C=us, O=juniper, CN=CRL1
    http://CA-1/CRL/juniper_us_crlfile.crl
  Use for key: Digital signature
```

To display the local certificate request, issue the **show security pki certificate-request** command:

```
user@R3> show security pki certificate-request
Certificate identifier: local-entrust3
  Issued to: router3.juniper.net
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

To display the local certificate, issue the **show security pki local-certificate** command:

```
user@R3> show security pki local-certificate
Certificate identifier: local-entrust3
  Issued to: router3.juniper.net, Issued by: juniper
  Validity:
    Not before: 2005 Nov 21st, 23:33:58 GMT
    Not after: 2008 Nov 22nd, 00:03:58 GMT
  Public key algorithm: rsaEncryption(1024 bits)
  Public key verification status: Passed
```

Router 4

On Router 4, issue a **ping** command to the **so-0/0/0** interface on Router 1 to send traffic across the IPSec tunnel.

```
user@R4> ping 10.1.12.2
PING 10.1.12.2 (10.1.12.2): 56 data bytes
64 bytes from 10.1.12.2: icmp_seq=0 ttl=254 time=1.350 ms
64 bytes from 10.1.12.2: icmp_seq=1 ttl=254 time=1.161 ms
64 bytes from 10.1.12.2: icmp_seq=2 ttl=254 time=1.124 ms
64 bytes from 10.1.12.2: icmp_seq=5 ttl=254 time=1.116 ms
^C
--- 10.1.12.2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 1.116/1.172/1.350/0.081 ms
```

The final way you can confirm that traffic travels over the IPSec tunnel is by issuing the **traceroute** command to the **so-0/0/0** interface on Router 1. Notice that the physical interface between Routers 2 and 3 is not referenced in the path; traffic enters the IPSec tunnel through the adaptive services IPSec inside interface on Router 3, passes through the loopback interface on Router 2, and ends at the **so-0/0/0** interface on Router 1.

```
user@R4> traceroute 10.1.12.2
traceroute to 10.1.12.2 (10.1.12.2), 30 hops max, 40 byte packets
 1  10.1.15.2 (10.1.15.2)  0.987 ms  0.630 ms  0.563 ms
 2  10.0.0.2 (10.0.0.2)  1.194 ms  1.058 ms  1.033 ms
 3  10.1.12.2 (10.1.12.2)  1.073 ms  0.949 ms  0.932 ms
```

For additional information on using digital certificates, see the *Junos Services Interfaces Configuration Guide* and the *Junos System Basics and Services Command Reference*.

Example: Dynamic Endpoint Tunneling Configuration

Figure 9: IPSec Dynamic Endpoint Tunneling Topology Diagram

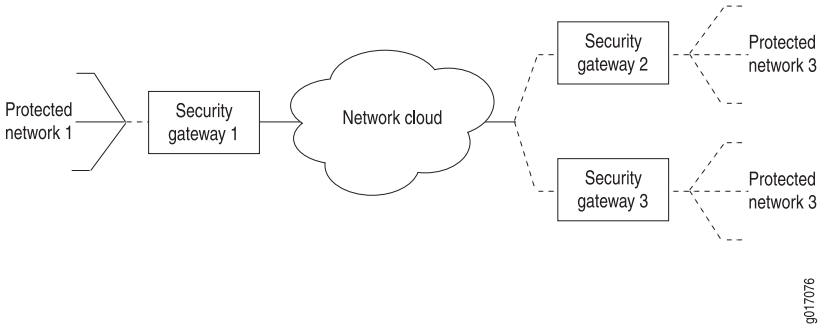


Figure 9 on page 108 shows a local network N-1 located behind security gateway SG-1. SG-1 is a Juniper Networks router terminating dynamic peer endpoints. The tunnel termination address on SG-1 is **10.7.7.2** and the local network address is **172.16.1.0/24**.

A remote peer router obtains addresses from an ISP pool and runs RFC-compliant IKE. Remote network N-2 has address **172.16.2.0/24** and is located behind security gateway SG-2 with tunnel termination address **10.7.7.1**.

On Router SG-1, configure an IKE access profile to accept proposals from SG-2. Apply the interface identifier from the access profile to the inside services interface and apply the IKE access profile itself to the IPSec next-hop style service set.

Router SG-1

```
[edit]
access {
    profile ike_access {
        client * { # Accepts proposals from specified peers that use the preshared key.
            ike {
                allowed-proxy-pair local 10.255.14.63/32 remote 10.255.14.64/32;
                pre-shared-key ascii-text "$9$1hoESeLxdgoGvWoGDif5IEc"; # SECRET-DATA
                interface-id test_id; # Apply this ID to the inside services interfaces.
            }
        }
    }
}
interfaces {
    fe-0/0/0 {
        description "Connection to the local network";
        unit 0 {
            family inet {
                address 172.16.1.1/24;
            }
        }
    }
    so-1/0/0 {
        description "Connection to SG-2";
        no-keepalives;
        encapsulation cisco-hdlc;
        unit 0 {
            family inet {
```

```
            address 10.7.7.2/30;
          }
        }
      }
      sp-3/3/0 {
        unit 0 {
          family inet;
        }
        unit 3 {
          dial-options {
            ipsec-interface-id test_id; # Accepts dynamic endpoint tunnels.
            shared;
          }
          service-domain inside;
        }
        unit 4 {
          family inet;
          service-domain outside;
        }
      }
    }
    services {
      service-set dynamic_nh_ss { # Create a next-hop service set
        next-hop-service { # for the dynamic endpoint tunnels.
          inside-service-interface sp-3/3/0.3;
          outside-service-interface sp-3/3/0.4;
        }
        ipsec-vpn-options {
          local-gateway 10.7.7.2;
          ike-access-profile ike_access; # Apply the IKE access profile here.
        }
      }
    }
```

Verifying Your Work

To verify proper operation of a dynamic endpoint tunnel configured on the AS PIC, use the following command:

show services ipsec-vpn ipsec security-associations (detail)

The following section shows output from this command used with the configuration example. The dynamically created rule **_junos_** appears in the output, as well as the establishment of the inbound and outbound dynamically created tunnels.

```
user@router> show services ipsec-vpn ipsec security-associations detail
Service set: dynamic_nh_ss

  Rule:  _junos_ , Term: tunnel4, Tunnel index: 4
  Local gateway: 10.7.7.2, Remote gateway: 10.7.7.1
  Local identity: ipv4(any:0,[0..3]=10.255.14.63)
  Remote identity: ipv4(any:0,[0..3]=10.255.14.64)

    Direction: inbound , SPI: 428111023, AUX-SPI: 0
     Mode: tunnel, Type: dynamic, State: Installed
    Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
    Soft lifetime: Expires in 27660 seconds
```

```
Hard lifetime: Expires in 27750 seconds
Anti-replay service: Enabled, Replay window size: 64

Direction: outbound , SPI: 4035429231, AUX-SPI: 0
 Mode: tunnel, Type: dynamic, State: Installed
Protocol: ESP, Authentication: hmac-sha1-96, Encryption: 3des-cbc
Soft lifetime: Expires in 27660 seconds
Hard lifetime: Expires in 27750 seconds
Anti-replay service: Enabled, Replay window size: 64
```

For More Information

For additional information about IPSec, see the following:

- *Junos System Basics Configuration Guide*
- *Junos Services Interfaces Configuration Guide*
- *Junos System Basics and Services Command Reference*
- RFC 1321, *The MD5 Message-Digest Algorithm*
- RFC 2104, *HMAC: Keyed-Hashing for Message Authentication*
- RFC 2401, *Security Architecture for the Internet Protocol*
- RFC 2402, *IP Authentication Header*
- RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*
- RFC 2406, *IP Encapsulating Security Payload (ESP)*
- RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*
- RFC 2409, *The Internet Key Exchange (IKE)*
- RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*
- RFC 3174, *US Secure Hash Algorithm 1 (SHA1)*
- RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*
- *Data Encryption Standard*, Federal Information Processing Standards (FIPS) Publication 46-3, **http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf** (includes information about DES and 3DES)
- *Descriptions of SHA-256, SHA-384, and SHA-512*, **http://csrc.nist.gov/cryptval/shs/sha256-384-512.pdf**
- Internet draft draft-eastlake-sha2-02.txt, *US Secure Hash Algorithms (SHA and HMAC-SHA)* (expires July 2006)

Revision History

January 2010—10.1 R1 Release. Merisha Wazna.

October 2009—10.0 R1 Release. Merisha Wazna.

July 2009—9.6 R1 Release. Merisha Wazna.

13 April 2009—9.5R1 Release. Roy Spencer.

15 January 2009—9.4R1 Release. Roy Spencer.

10 October 2008—9.3R1 Release. Roy Spencer.

10 July 2008—9.2R1 Release. Roy Spencer.

10 April 2008—9.1R1 Release. Roy Spencer.

1 February 2008—9.0R1 Release. Fawn Damitio.

27 March 2007—Added support for IPSec over OSPFv2 links. 8.2R1 Release. Fawn Damitio.

12 January 2007—Added support for M120 routers. 8.2R1 Release. Fawn Damitio.

15 September 2006—Added support for certificate revocation lists (CRLs), and IPSec IKE in routing instances, 8.1R1 Release. Ines Salazar and Richard Hendricks.

29 June 2006—8.0R1 Release. Richard Hendricks.

27 March 2006—Added support for AES encryption and SHA-256 authentication on J Series Services Routers and AS PICs installed in M Series routers, and IPv6-based IPSec for AS PICs installed in M Series and T Series routers, 7.6R1 Release. Richard Hendricks.

9 January 2006—Added support for digital certificates on J Series Services Routers and AS PICs installed in M Series and T Series routers, and support for the IPSec Monitoring MIB, 7.5R1 Release. Richard Hendricks.

14 September 2005—Added support for dynamic endpoint tunneling and configuring multiple routed tunnels in a single next-hop service set, 7.4R1 Release. Richard Hendricks.

13 June 2005—7.3R1 Release. Richard Hendricks.

5 April 2005—Added support for transport mode IPSec in Routing Engines running OSPFv3 and support for the AS II FIPS PIC, 7.2R1 Release. Richard Hendricks.

2 February 2005—Document converted to Feature Guide format, thoroughly revised, and enhanced with updated examples, Junos OS Release 7.1R1. Richard Hendricks.

07 March 2002—Initial Quick Start Guide written. Tony Sinopoli.

PART 2

Index

Index