VPN Acceleration Module

Last Updated: December 3, 2012

VPN Acceleration Module (VAM) supports Data Encryption Standard (DES) or Triple DES (3DES) IPsec encryption at a rate greater than full-duplex DS-3 line rate (up to 145 Mbps) for site-to-site VPNs such as intranets and extranets. VAM also supports up to 5000 encrypted tunnels for mixed VPN environments that have both site-to-site and remote access VPN requirements. VAM integrates hardware-assisted Rivest, Shamir, and Adelman (RSA) and IP Payload Compression Protocol (IPPCP) layer 3 compression to accelerate RSA processing, thereby enhancing tunnel setup and improving overall VPN initialization. In environments where bandwidth is costly, VAM provides hardware-based IPPCP Lempel-Ziv-Stac (LZS) processing to compress network traffic before it is encrypted and sent over pay-per-byte WAN connections.

- Finding Feature Information, page 1
- Prerequisites for VPN Acceleration Module, page 1
- Information about VPN Acceleration Module (VAM), page 2
- How To Configure VPN Acceleration Module (VAM), page 5
- Configuration Examples for VPN Acceleration, page 17
- Additional References for VPN Acceleration Module, page 18
- Feature Information for VPN Acceleration Module, page 19
- Glossary, page 21

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Prerequisites for VPN Acceleration Module

You must configure IPSec and IKE on the router and a crypto map to all interfaces that require encryption service from the VPN Acceleration Module.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information about VPN Acceleration Module (VAM)

- VPN Acceleration Module (VAM) Overview, page 2
- Benefits, page 2

VPN Acceleration Module (VAM) Overview

The VPN Acceleration Module (VAM) is a single-width acceleration module. It provides high-performance, hardware-assisted tunneling and encryption services suitable for VPN remote access, site-to-site intranet, and extranet applications. It also provides platform scalability and security while working with all services necessary for successful VPN deployments—security, quality of service (QoS), firewall and intrusion detection, service-level validation, and management. The VAM off-loads IPsec processing from the main processor, thus freeing resources on the processor engines for other tasks.

The VAM provides hardware-accelerated support for the following multiple encryption functions:

- 56-bit Data Encryption Standard (DES) standard mode: Cipher Block Chaining (CBC)
- 3-Key Triple DES (168-bit)
- Secure Hash Algorithm (SHA)-1 and Message Digest 5 (MD5)
- Rivest, Shamir, Adelman (RSA) public-key algorithm
- Diffie-Hellman key exchange RC4-40

Benefits

The VAM provides the following benefits:

- 10 tunnels per second
- The following number of tunnels based on the corresponding memory of the NPE:
  - 800 tunnels for 64 MB
  - 1600 tunnels for 128 MB
  - 3200 tunnels for 256 MB
  - 5000 tunnels for 512 MB
- RSA encryption
- Accelerated Crypto performance
- Accelerated Internet Key Exchange (IKE)
- Certificate support for automatic authentication using digital certificates
- Dual VAM support

Note Support for dual VAMs is available on a Cisco 7200 series router with an NPE-G1, on Cisco IOS Release 12.2(15)T, 12.1(14)E, and 12.3 Mainline.

- Encryption services to any port adapter installed in the router. The interface on the port adapter must be configured with a crypto map to support IPSec.
- Full-duplex data transmission of over 100 Mbps with various encryption and compression schemes for 300 byte packages
- Hardware-based IPPCP LZS compression

- Network traffic compression that reduces bandwidth utilization
- Online Insertion and Removal (OIR)
- QoS, multiprotocol, and multicast feature interoperation
- Support for full Layer 3 routing, such as Enhanced Interior Gateway Routing Protocol (EIGRP), Open Shortest Path First (OSPF), and Border Gateway Protocol (BGP) across the IPSec VPN
- Up to 145 Mbps throughput using 3DES
- VPN initialization improvements

Performance Results for Single VAM

The following two tables provide performance results for a single VAM on a Cisco 7206VXR with an NPE-G1 processor, an onboard GE, and FE port adapters in slots 3 and 4.

clear_packet _size	crypto_packet_size	out_packet_size
64	96	114
300	336	354
1400	1432	1450
Mixed packet size - 344	378	396

pkt_size (bytes)	# of tunnels	measured_pps (pps)	meas_clear_ndr (Mbps)	meas_crypto_ndr (Mbps)	meas_out_ndr (Mbps)
64	4	65,224	33.39	50.09	59.48
	500	41,888	21.44	32.17	38.20
	1,000	40,480	20.73	31.09	36.92
	5,000	39,408	20.18	30.27	35.94
300	4	38,032	91.28	102.23	107.71
	500	37,184	89.24	99.95	105.31
	1,000	36,064	86.55	96.94	102.13
	5,000	36,016	86.44	96.81	101.99
1400	4	9,984	111.82	114.38	115.81
	500	9,848	110.29	112.82	114.24
	1,000	9,648	108.06	110.53	111.92
	5,000	9,616	107.70	110.16	111.55
Mixed packet size	4	31,472	86.61	95.17	99.70
	500	31,056	85.47	93.91	98.39

pkt_size (bytes)	# of tunnels	measured_pps (pps)	meas_clear_ndr (Mbps)	meas_crypto_ndr (Mbps)	meas_out_ndr (Mbps)
	1,000	30,128	82.91	91.11	95.45
	5,000	29,264	80.53	88.49	92.71

Performance Results for Dual VAMs

The following two tables provide performance results for dual VAMs on a Cisco 7206VXR with an NPE-G1 processor, an onboard GE, and FE port adapters in slots 3 and 4.

clear_packet _size	crypto_packet_size	out_packet_size
64	96	114
300	336	354
1400	1432	1450
Mixed packet size - 344	378	396

pkt_size (bytes)	# of tunnels	measured_pps (pps)	meas_clear_ndr (Mbps)	meas_crypto_ndr (Mbps)	meas_out_ndr (Mbps)
64	4	135,544	69.40	104.10	123.61
	500	61,520	31.50	47.25	56.11
	1,000	56,928	29.15	43.72	51.92
	5,000	43,744	22.40	33.60	39.89
300	4	71,336	171.21	191.75	202.02
	500	60,416	145.00	162.40	171.10
	1,000	56,016	134.44	150.57	158.64
	5,000	42,496	101.99	114.23	120.35
1400	4	18,736	209.84	214.64	217.34
	500	18,424	206.35	211.07	213.72
	1000	18,352	205.54	210.24	212.88
	5,000	18,352	205.54	210.24	212.88
Mixed packet size	4	60,416	166.26	182.70	191.40
	500	57,888	159.31	175.05	183.40
	1,000	55,488	152.70	167.80	175.79

pkt_size (bytes)	# of tunnels	measured_pps (pps)	meas_clear_ndr (Mbps)	meas_crypto_ndr (Mbps)	meas_out_ndr (Mbps)
	5,000	34,272	94.32	103.64	108.57

How To Configure VPN Acceleration Module (VAM)

On power up if the enabled LED is on, the VAM is fully functional and does not require any configuration commands. However, for the VAM to provide encryption services, you must complete the following tasks:

- Creating IKE Policies, page 5
- Configuring IPsec, page 8
- Troubleshooting Tips, page 15
- Monitoring and Maintaining the VPN Acceleration Module, page 16

Creating IKE Policies

The following restrictions apply if you are configuring an AES IKE policy:

- Your device must support IPsec and long keys (the “k9” subsystem).
- AES cannot encrypt IPsec and IKE traffic if an acceleration card is present.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp policy** *priority*
4. **encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**}
5. **hash** {**sha** | **sha256** | **sha384** | **md5**}
6. **authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
7. **group** {**1** | **2** | **5** | **14** | **15** | **16** | **19** | **20** | **24**}
8. **lifetime** *seconds*
9. **exit**
10. **exit**
11. **show crypto isakmp policy**
12. Repeat these steps for each policy you want to create.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 10`	Defines an IKE policy and enters config-isakmp configuration mode. • *priority* —Uniquely identifies the IKE policy and assigns a priority to the policy. Valid values: 1 to 10,000; 1 is the highest priority.

Step 4	**encryption** {**des** | **3des** | **aes** | **aes 192** | **aes 256**}
	Example:
	`Router(config-isakmp)# encryption aes 256`
		Specifies the encryption algorithm.
		• By default, the **des** keyword is used.
		◦ **des**—56-bit DES-CBC (No longer recommended. AES is the recommended encryption algorithm)
		◦ **3des**—168-bit DES (No longer recommended. AES is the recommended encryption algorithm)
		◦ **aes**—128-bit AES
		◦ **aes 192**—192-bit AES
		◦ **aes 256**—256-bit AES

Step 5	**hash** {**sha** | **sha256** | **sha384** | **md5**}
	Example:
	`Router(config-isakmp)# hash sha`
		Specifies the hash algorithm.
		• By default, SHA-1 (**sha**) is used.
		• The **sha256** keyword specifies SHA-2 family 256-bit (HMAC variant) as the hash algorithm.
		• The **sha384** keyword specifies SHA-2 family 384-bit (HMAC variant) as the hash algorithm.
		• The **md5** keyword specifies MD5 (HMAC variant) as the hash algorithm. (No longer recommended. SHA-256 is the recommended replacement.)

Step 6	**authentication** {**rsa-sig** | **rsa-encr** | **pre-share**}
	Example:
	`Router(config-isakmp)# authentication pre-share`
		Specifies the authentication method.
		• By default, RSA signatures are used.
		◦ **rsa-sig**—RSA signatures require that you configure your peer routers to obtain certificates from a CA.
		◦ **rsa-encr**—RSA encrypted nonces require that you ensure each peer has the other peer’s RSA public keys.
		◦ **pre-share**—Preshared keys require that you separately configure these preshared keys.

	Command or Action	Purpose

Step 7	**group** {**1** | **2** | **5** | **14** | **15** | **16** | **19** | **20** | **24**}
	Example:
	`Router(config-isakmp)# group 14`
		Specifies the Diffie-Hellman (DH) group identifier.
		• By default, DH group 1 is used.
		◦ **1**—768-bit DH (No longer recommended.)
		◦ **2**—1024-bit DH (No longer recommended)
		◦ **5**—1536-bit DH (No longer recommended)
		◦ **14**—Specifies the 2048-bit DH group.
		◦ **15**—Specifies the 3072-bit DH group.
		◦ **16**—Specifies the 4096-bit DH group.
		◦ **19**—Specifies the 256-bit elliptic curve DH (ECDH) group.
		◦ **20**—Specifies the 384-bit ECDH group.
		◦ **24**—Specifies the 2048-bit DH/DSA group.
		The group chosen must be strong enough (have enough bits) to protect the IPsec keys during negotiation. A generally accepted guideline recommends the use of a 2048-bit group after 2013 (until 2030). Group 14 or higher (where possible) can be selected to meet this guideline. Even if a longer-lived security method is needed, the use of Elliptic Curve Cryptography is recommended, but group 15 and group 16 can also be considered.
		The ISAKMP group and the IPsec perfect forward secrecy (PFS) group should be the same if PFS is used. If PFS is not used, a group is not configured in the IPsec crypto map.

Step 8	**lifetime** *seconds* **Example:** `Router(config-isakmp)# lifetime 180`	Specifies the lifetime of the IKE SA. • *seconds*—Time, in seconds, before each SA expires. Valid values: 60 to 86,400; default value: 86,400. **Note** The shorter the lifetime (up to a point), the more secure your IKE negotiations will be. However, with longer lifetimes, future IPsec SAs can be set up more quickly.
Step 9	**exit** **Example:** `Router(config-isakmp)# exit`	Exits config-isakmp configuration mode.
Step 10	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

	Command or Action	Purpose
Step 11	**show crypto isakmp policy** **Example:** `Router# show crypto isakmp policy`	(Optional) Displays all existing IKE policies.
Step 12	Repeat these steps for each policy you want to create.	—

Examples

The following sample output from the **show crypto isakmp policy** command displays a warning message after a user tries to configure an IKE encryption method that the hardware does not support:

```
Router# show crypto isakmp policy
Protection suite of priority 1
        encryption algorithm:  AES - Advanced Encryption Standard (256 bit keys).
WARNING:encryption hardware does not support the configured
encryption method for ISAKMP policy 1
        hash algorithm:        Secure Hash Standard 2 (256-bit)
        authentication method: Pre-Shared Key
        Diffie-Hellman group:  #14 (2048 bit)
        lifetime:              3600 seconds, no volume limit
```

Configuring IPsec

After you have completed IKE configuration, configure IPsec at each participating IPsec peer. This section contains basic steps to configure IPsec.

- Creating Crypto Access Lists, page 8
- Configuring Transform Sets for IKEv1, page 9
- Creating Static Crypto Maps, page 11
- Verifying the Configuration, page 13

Creating Crypto Access Lists

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
    - **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**]
    - **ip access-list extended** *name*
4. Repeat Step 3 for each crypto access list you want to create.

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	Do one of the following:
	• **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**]
	• **ip access-list extended** *name*
	Example:
	`Device(config)# access-list 100 permit ip 10.0.68.0 0.0.0.255 10.1.1.0 0.0.0.255`
	Example:
	`Device(config)# ip access-list extended vpn-tunnel`
		Specifies conditions to determine which IP packets are protected.
		• You specify conditions using an IP access list designated by either a number or a name. The **access-list** command designates a numbered extended access list; the **ip access-list extended** command designates a named access list.
		• Enable or disable crypto for traffic that matches these conditions.
		Tip Cisco recommends that you configure “mirror image” crypto access lists for use by IPsec and that you avoid using the **any** keyword.

Step 4	Repeat Step 3 for each crypto access list you want to create.	—

Configuring Transform Sets for IKEv1

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]]
4. **mode** [**tunnel** | **transport**]
5. **end**
6. **clear crypto sa** [**peer** {*ip-address* | *peer-name*} | **sa map** *map-name* | **sa entry** *destination-address protocol spi*]
7. **show crypto ipsec transform-set** [**tag** *transform-set-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform1* [*transform2* [*transform3*]] **Example:** `Device(config)# crypto ipsec transform-set aesset esp-aes 256 esp-sha-hmac`	Defines a transform set and enters crypto transform configuration mode. • There are complex rules defining the entries that you can use for transform arguments. These rules are explained in the command description for the **crypto ipsec transform-set** command, and the table in "About Transform Sets" section provides a list of allowed transform combinations.
Step 4	**mode** [**tunnel** | **transport**] **Example:** `Device(cfg-crypto-tran)# mode transport`	(Optional) Changes the mode associated with the transform set. • The mode setting is applicable only to traffic whose source and destination addresses are the IPsec peer addresses; it is ignored for all other traffic. (All other traffic is in tunnel mode only.)

Step 5	**end**
	Example:
	`Device(cfg-crypto-tran)# end`
		Exits crypto transform configuration mode and enters privileged EXEC mode.

Step 6	**clear crypto sa** [**peer** {*ip-address* | *peer-name*} | **sa map** *map-name* | **sa entry** *destination-address protocol spi*] **Example:** `Device# clear crypto sa`	(Optional) Clears existing IPsec security associations so that any changes to a transform set takes effect on subsequently established security associations. Manually established SAs are reestablished immediately. • Using the **clear crypto sa** command without parameters clears out the full SA database, which clears out active security sessions. • You may also specify the **peer**, **map**, or **entry** keywords to clear out only a subset of the SA database.
Step 7	**show crypto ipsec transform-set** [**tag** *transform-set-name*] **Example:** `Device# show crypto ipsec transform-set`	(Optional) Displays the configured transform sets.

Creating Static Crypto Maps

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

Perform this task to create crypto map entries that use IKE to establish SAs. To create IPv6 crypto map entries, you must use the **ipv6** keyword with the **crypto map** command. For IPv4 crypto maps, use the **crypto map** command without the **ipv6** keyword.

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the *Next Generation Encryption* (NGE) white paper.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** [**ipv6**] *map-name seq-num* [**ipsec-isakmp**]
4. **match address** *access-list-id*
5. **set peer** {*hostname* | *ip-address*}
6. **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]
7. **set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes* | **kilobytes disable**}
8. **set security-association level per-host**
9. **set pfs** [**group1** | **group14** | **group15** | **group16** | **group19** | **group2** | **group20** | **group24** | **group5**]
10. **end**
11. **show crypto map** [**interface** *interface* | **tag** *map-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto map** [**ipv6**] *map-name seq-num* [**ipsec-isakmp**] **Example:** `Device(config)# crypto map static-map 1 ipsec-isakmp`	Creates or modifies a crypto map entry, and enters crypto map configuration mode. • For IPv4 crypto maps, use the command without the **ipv6** keyword.
Step 4	**match address** *access-list-id* **Example:** `Device(config-crypto-m)# match address vpn-tunnel`	Names an extended access list. • This access list determines the traffic that should be protected by IPsec and the traffic that should not be protected by IPsec security in the context of this crypto map entry.

Step 5	**set peer** {*hostname* | *ip-address*}
	Example:
	`Device(config-crypto-m)# set-peer 192.168.101.1`
		Specifies a remote IPsec peer—the peer to which IPsec protected traffic can be forwarded.
		• Repeat for multiple remote peers.

Step 6	**set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*] **Example:** `Device(config-crypto-m)# set transform-set aesset`	Specifies the transform sets that are allowed for this crypto map entry. • List multiple transform sets in the order of priority (highest priority first).

Step 7	**set security-association lifetime** {**seconds** *seconds* | **kilobytes** *kilobytes* | **kilobytes disable**}
	Example:
	`Device (config-crypto-m)# set security-association lifetime seconds 2700`
		(Optional) Specifies a SA lifetime for the crypto map entry.
		• By default, the SAs of the crypto map are negotiated according to the global lifetimes, which can be disabled.

Step 8	**set security-association level per-host** **Example:** `Device(config-crypto-m)# set security-association level per-host`	(Optional) Specifies that separate SAs should be established for each source and destination host pair. • By default, a single IPsec “tunnel” can carry traffic for multiple source hosts and multiple destination hosts. **Caution** Use this command with care because multiple streams between given subnets can rapidly consume resources.

	Command or Action	Purpose
Step 9	**set pfs** [**group1** | **group14** | **group15** | **group16** | **group19** | **group2** | **group20** | **group24** | **group5**] **Example:** `Device(config-crypto-m)# set pfs group14`	(Optional) Specifies that IPsec either should ask for password forward secrecy (PFS) when requesting new SAs for this crypto map entry or should demand PFS in requests received from the IPsec peer. • Group 1 specifies the 768-bit Diffie-Hellman (DH) identifier (default). (No longer recommended). • Group 2 specifies the 1024-bit DH identifier. (No longer recommended). • Group 5 specifies the 1536-bit DH identifier. (No longer recommended) • Group 14 specifies the 2048-bit DH identifier. • Group 15 specifies the 3072-bit DH identifier. • Group 16 specifies the 4096-bit DH identifier. • Group 19 specifies the 256-bit elliptic curve DH (ECDH) identifier. • Group 20 specifies the 384-bit ECDH identifier. • Group 24 specifies the 2048-bit DH/DSA identifier • By default, PFS is not requested. If no group is specified with this command, group 1 is used as the default.

Step 10	**end**
	Example:
	`Device(config-crypto-m)# end`
		Exits crypto map configuration mode and returns to privileged EXEC mode.

Step 11	**show crypto map** [**interface** *interface* | **tag** *map-name*] **Example:** `Device# show crypto map`	Displays your crypto map configuration.

Verifying the Configuration

SUMMARY STEPS

1. **show crypto ipsec transform-set**
2. **show crypto map** [**interface** *interface* | **tag** *map-name*]
3. **show crypto ipsec sa** [**map** *map-name* | **address** | **identity** | **detail** | **interface**]

DETAILED STEPS

Step 1 **show crypto ipsec transform-set**

Example:

```
Device# show crypto ipsec transform-set

Transform set combined-des-md5: {esp-des esp-md5-hmac}
   will negotiate = {Tunnel,},
Transform set t1: {esp-des esp-md5-hmac}
   will negotiate = {Tunnel,},
Transform set t100: {ah-sha-hmac}
   will negotiate = {Transport,},
Transform set t2: {ah-sha-hmac}
   will negotiate = {Tunnel,},
   {esp-des}
   will negotiate = {Tunnel,},
```

Displays the transform set configuration.

Step 2 **show crypto map** [**interface** *interface* | **tag** *map-name*]

Example:

```
Device# show crypto map

Crypto Map “abc” 10 ipsec-isakmp
        Peer = 172.21.114.67
        Extended IP access list 141
            access-list 141 permit ip
                source: addr = 172.21.114.123/0.0.0.0
                dest:   addr = 172.21.114.67/0.0.0.0
        Current peer: 172.21.114.67
        Security-association lifetime: 4608000 kilobytes/120 seconds
        PFS (Y/N): N
        Transform sets={t1,}
```

Displays the crypto map configuration.

Step 3 **show crypto ipsec sa** [**map** *map-name* | **address** | **identity** | **detail** | **interface**]

Example:

```
Device# show crypto map ipsec sa interface

    Crypto map tag: abc, local addr. 172.21.114.123
   local ident (addr/mask/prot/port): (172.21.114.123/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (172.21.114.67/255.255.255.255/0/0)
    current_peer: 172.21.114.67
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify 10
    #send errors 10, #recv errors 0
     local crypto endpt.: 172.21.114.123, remote crypto endpt.: 172.21.114.67
     path mtu 1500, media mtu 1500
     current outbound spi: 20890A6F
     inbound esp sas:
      spi: 0x257A1039(628756537)
        transform: esp-des esp-md5-hmac,
        in use settings ={Tunnel,}
        slot: 0, conn id: 26, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y
     inbound ah sas:
     outbound esp sas:
      spi: 0x20890A6F(545852015)
        transform: esp-des esp-md5-hmac,
        in use settings ={Tunnel,}
        slot: 0, conn id: 27, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y
     outbound ah sas:
interface: Tunnel0
   Crypto map tag: abc, local addr. 172.21.114.123
   local ident (addr/mask/prot/port): (172.21.114.123/255.255.255.255/0/0)
   remote ident (addr/mask/prot/port): (172.21.114.67/255.255.255.255/0/0)
   current_peer: 172.21.114.67
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 10, #pkts encrypt: 10, #pkts digest 10
    #pkts decaps: 10, #pkts decrypt: 10, #pkts verify 10
    #send errors 10, #recv errors 0
     local crypto endpt.: 172.21.114.123, remote crypto endpt.: 172.21.114.67
     path mtu 1500, media mtu 1500
     current outbound spi: 20890A6F
     inbound esp sas:
      spi: 0x257A1039(628756537)
        transform: esp-des esp-md5-hmac,
        in use settings ={Tunnel,}
        slot: 0, conn id: 26, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y
     inbound ah sas:
     outbound esp sas:
      spi: 0x20890A6F(545852015)
        transform: esp-des esp-md5-hmac,
        in use settings ={Tunnel,}
        slot: 0, conn id: 27, crypto map: router-alice
        sa timing: remaining key lifetime (k/sec): (4607999/90)
        IV size: 8 bytes
        replay detection support: Y
     outbound ah sas:
```

Displays information about IPsec security associations.

Troubleshooting Tips

To verify that Cisco IOS software has recognized VAM, enter the **show diag** command and check the output. For example, when the router has the VAM in slot 1, the following output appears:

```
Router# show diag 1
    Slot 1:
            VAM Encryption/Compression engine. Port adapter
            Port adapter is analyzed
            Port adapter insertion time 00:04:45 ago
            EEPROM contents at hardware discovery:
            Hardware Revision        :1.0
            PCB Serial Number        :15485660
            Part Number              :73-5953-04
            Board Revision           :
            RMA Test History         :00
            RMA Number               :0-0-0-0
            RMA History              :00
            Deviation Number         :0-0
            Product Number           :CLEO
            Top Assy. Part Number    :800-10496-04
            CLEI Code                :
            EEPROM format version 4
            EEPROM contents (hex):
              0x00:04 FF 40 02 8A 41 01 00 C1 8B 31 35 34 38 35 36
              0x10:36 30 00 00 00 82 49 17 41 04 42 FF FF 03 00 81
              0x20:00 00 00 00 04 00 80 00 00 00 00 CB 94 43 4C 45
              0x30:4F 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
              0x40:20 C0 46 03 20 00 29 00 04 C6 8A FF FF FF FF FF
```

```
0x50:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x60:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
0x70:FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

To see if the VAM is currently processing crypto packets, enter the **show pas vam interface** command. The following is sample output:

```
Router# show pas vam interface


Interface VAM 1/1 :
        ds:0x632770C8        idb:0x62813728
        Statistics of packets and bytes that through this interface:
            18 packets in                     18 packets out
          2268 bytes in                     2268 bytes out
             0 paks/sec in                     0 paks/sec out
             0 Kbits/sec in                    0 Kbits/sec out
            83 commands out                   83 commands acknowledged
        ppq_full_err   :0            ppq_rx_err       :0
        cmdq_full_err  :0            cmdq_rx_err      :0
        no_buffer      :0            fallback         :0
        dst_overflow   :0            nr_overflow      :0
        sess_expired   :0            pkt_fragmented   :0
        out_of_mem     :0            access_denied   :0
        invalid_fc     :0            invalid_param    :0
        invalid_handle :0            output_overrun   :0
        input_underrun :0            input_overrun    :0
        key_invalid    :0            packet_invalid   :0
        decrypt_failed :0            verify_failed    :0
        attr_invalid   :0            attr_val_invalid :0
        attr_missing   :0            obj_not_wrap     :0
        bad_imp_hash   :0            cant_fragment    :0
        out_of_handles :0            compr_cancelled  :0
        rng_st_fail    :0            other_errors     :0
        633 seconds since last clear of counters
```

When the VAM processes packets, the “packet in” and “packet out” counters change. Counter “packets out” represents the number of packets directed to the VAM. Counter “packets in” represents the number of packets received from the VAM.

Note In versions prior to Cisco IOS Release 12.2(5)T and Cisco IOS Release 12.1(10)E, upon reboot trap configurations are lost and need to be re-entered.

Monitoring and Maintaining the VPN Acceleration Module

Use the commands below to monitor and maintain the VPN Acceleration Module:

Command	Purpose
`Router#` **show pas isa interface**	Displays the ISA interface configuration.
`Router#` **show pas isa controller**	Displays the ISA controller configuration.
`Router#` **show pas vam interface**	Verifies the VAM is currently processing crypto packets.
`Router#` **show pas vam controller**	Displays the VAM controller configuration.

Command	Purpose
`Router#` **Show version**	Displays integrated service adapter as part of the interfaces.

Configuration Examples for VPN Acceleration

- Example: Configuring IKE Policies, page 17
- Example: Configuring IPsec Configuration, page 17

Example: Configuring IKE Policies

In the following example, two IKE policies are created, with policy 15 as the highest priority, policy 20 as the next priority, and the existing default priority as the lowest priority. It also creates a preshared key to be used with policy 20 with the remote peer whose IP address is 192.168.224.33.

```
crypto isakmp policy 15
 encryption 3des
 hash md5
 authentication rsa-sig
 group 2
 lifetime 5000
crypto isakmp policy 20
 authentication pre-share
 lifetime 10000
crypto isakmp key 1234567890 address 192.168.224.33
```

Example: Configuring IPsec Configuration

The following example shows a minimal IPsec configuration where the security associations will be established via IKE:

An IPsec access list defines which traffic to protect:

```
access-list 101 permit ip 10.0.0.0 0.0.0.255 10.2.2.0 0.0.0.255
```

A transform set defines how the traffic will be protected. In this example, transform set "myset1" uses DES encryption and SHA for data packet authentication:

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

Another transform set example is "myset2," which uses Triple DES encryption and MD5 (HMAC variant) for data packet authentication:

```
crypto ipsec transform-set myset2 esp-3des esp-md5-hmac
```

A crypto map joins together the IPsec access list and transform set and specifies where the protected traffic is sent (the remote IPsec peer):

```
crypto map toRemoteSite 10 ipsec-isakmp
 match address 101
 set transform-set myset2
 set peer 10.2.2.5
```

The crypto map is applied to an interface:

```
interface Serial0
 ip address 10.0.0.2
 crypto map toRemoteSite
```

Note In this example, IKE must be enabled.

Additional References for VPN Acceleration Module

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Security commands	- *Cisco IOS Security Command Reference Commands A to C* - *Cisco IOS Security Command Reference Commands D to L* - *Cisco IOS Security Command Reference Commands M to R* - *Cisco IOS Security Command Reference Commands S to Z*
IPsec configuration	*Configuring IPsec*
IKE configuration	*Configuring IKE*
VAM installation and configuration tasks	*VAM Installation and Configuration Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2393	*IP Payload Compression Protocol (IPComp)*
RFC 2395	*IP Payload Compression Using LZS*
RFCs 2401 to 2411	*IPsec—IKE*
RFC 2451	*The ESP CBC-Mode Cipher Algorithms*

Standard/RFC	Title
IPSec/IKE: RFCs 2401-2411, 2451	

MIBs

MIB	MIBs Link
• CISCO-IPSEC-FLOW-MONITOR-MIB • CISCO-IPSEC-MIB • CISCO-IPSEC-POLICY-MAP-MIB	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for VPN Acceleration Module

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for VPN Acceleration Module

Feature Name	Releases	Feature Information
VPN Acceleration Module (VAM)	12.1(9)E 12.1(14)E 12.2(9)YE 12.2(13)T 12.2(15)T 12.3(1)Mainline 12.2(14)SU	VPN Acceleration Module (VAM) supports Data Encryption Standard (DES) or Triple DES (3DES) IPsec encryption at a rate greater than full-duplex DS-3 line rate (up to 145 Mbps) for site-to-site VPNs such as intranets and extranets. VAM also supports up to 5000 encrypted tunnels for mixed VPN environments that have both site-to-site and remote access VPN requirements. VAM integrates hardware-assisted Rivest, Shamir, and Adelman (RSA) and IP Payload Compression Protocol (IPPCP) layer 3 compression to accelerate RSA processing, thereby enhancing tunnel setup and improving overall VPN initialization. In environments where bandwidth is costly, VAM provides hardware-based IPPCP Lempel-Ziv-Stac (LZS) processing to compress network traffic before it is encrypted and sent over pay-per-byte WAN connections. In 12.1(9)E, this feature was introduced on the Cisco 7200 series routers on NPE-225, NPE-400, and NSE-1. In 12.1(14)E, this feature was integrated into Cisco IOS Release 12.1(14)E and support for dual VAMs[1] on the Cisco 7200 series with NPE-G1 was added. In 12.2(9)YE, support for this feature was added to the Cisco 7401ASR router[2]. The following commands were introduced or modified:**crypto engine sw ipsec, show pas vam**

[1] Support for dual VAMs is available on a Cisco 7200 series router with NPE-G1 on Cisco IOS Release 12.2(15)T, 12.1(14)E, and 12.3 Mainline only.

[2] The Cisco 7401ASR router is no longer sold.

Feature Name	Releases	Feature Information
		controller, show pas vam interface.

Glossary

IKE—Internet Key Exchange. IKE establishes a shared security policy and authenticates keys for services (such as IPSec) that require keys. Before any IPSec traffic can be passed, each router/firewall/host must verify the identity of its peer. This can be done by manually entering preshared keys into both hosts or by a CA service.

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

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.