Reference Guide to Implementing Crypto and QoS

Contents

- **Introduction**
- **Prerequisites**
  - Requirements
  - Components Used
  - Conventions
- **IPSec Protocols**
- **AH and ESP**
- **Use GRE Tunnels with IPSec**
- **Classify Packets**
- **Sample Configuration**
  - Input Policy
  - Output Policy
- **Restrictions and Related Issues**
  - QoS and Anti-Replay Protection
  - NBAR
  - Double Accounting
  - Software Encryption and Fast Switching/CEF
  - Legacy Priority Queuing and QoS PreClassify
  - Hardware Encryption and QoS
- **Cisco Support Community - Featured Conversations**
- **Related Information**

Introduction

As VPNs grow to include data, voice, and video traffic, the different types of traffic need to be handled differently in the network. Quality of service (QoS) and bandwidth management features allow a VPN to deliver high transmission quality for time-sensitive applications such as voice and video. Each packet is tagged to identify the priority and time sensitivity of its payload, and traffic is sorted and routed based on its delivery priority. Cisco VPN solutions support a wide range of QoS features.

This document is designed to serve as a single reference for users who configure Cisco IOS® encryption and QoS features on the same network or set of routers. You will see basic configurations of both input and output QoS policies in the presence of IP Security (IPSec) and generic routing encapsulation (GRE) tunnels. This document helps you to understand the configuration tasks. It also provides information on restrictions and known issues, to ensure optimal performance and successful implementation of enhanced IP services using Cisco routers.

Prerequisites

Requirements

Readers of this document should have knowledge of these topics:

- IPSec technology

For a more exhaustive document on IPSec, refer to An Introduction to IP Security (IPSec) Encryption.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

IPSec Protocols

A detailed discussion of the IPSec protocols is beyond the scope of this document. However, an overview is provided in this section. See the Related Information section of this document for further IPSec resources.

IPSec defines a network-layer authentication and encryption model. It consists of an encryption key exchange to build a secure connection, and authentication and encryption protocols that the two peers negotiate and then use throughout the lifetime of the encrypted connection.

Internet Security Association and Key Management Protocol (ISAKMP) negotiates encryption policy and provides a common framework to generate the keys shared by IPSec peers. The result of ISAKMP negotiations is a Security Association (SA). This example output of the **show crypto isakmp policy** command illustrates the parameters used during negotiation of a SA:

```
P5R0#show crypto isakmp policy
Protection suite of priority 100
        encryption algorithm:   DES - Data Encryption Standard (56 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Pre-Shared Key

!--- Supports pre-shared keys or a public/private


!--- key mechanism such as RSA.

        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit

!--- Lifetime can be based on time or on the  number of transmitted bytes.

Default protection suite
        encryption algorithm:   DES - Data Encryption Standard (56 bit keys).
        hash algorithm:         Secure Hash Standard
        authentication method:  Rivest-Shamir-Adleman Signature
        Diffie-Hellman group:   #1 (768 bit)
        lifetime:               86400 seconds, no volume limit
```

ISAKMP negotiations result in an encrypted connection (and SA) through which the peers negotiate additional IPSec parameters. This includes the actual protocols used to protect the IP datagrams. Specifically, the two encryption peers negotiate the parameters in this table.

Parameter	Options

IPSec Protocol Set	• Authentication Header (AH) • Encapsulating Security Payload (ESP) For most applications, either AH or ESP is sufficient.
Mode	• Tunnel mode - Inserts a new, encapsulating IP header while it retains and encrypts the original IP header. • Transport mode - Retains the original IP header, but changes the protocol field to a value of 51 or 50 to reflect the AH or ESP header, respectively. While in the negotiation phase, ISAKMP automatically "falls back" to tunnel mode if transport mode is desired, but cannot be established. **Note:** RFC 2401 uses the terms "outer" or "encapsulating" header to describe the new IP header with tunnel mode and "inner" header to describe the original IP header. Quote from the RFC: "The outer IP header Source Address and Destination Address identify the "endpoints" of the tunnel (the encapsulator and decapsulator). The inner IP header Source Address and Destination Addresses identify the original sender and recipient of the datagram, (from the perspective of this tunnel), respectively." These terms are used in the remainder of this document.
Transform Set	• AH for authentication only. • ESP for authentication and encryption.

The output from the **show crypto ipsec sa address** command illustrates the IPSec SA, each of which is identified by a Security Parameter Index (SPI). For example, the connection identified with a SPI of 0x21A85375 (564679541) uses the MD5-HMAC algorithm for AH and DES for ESP.

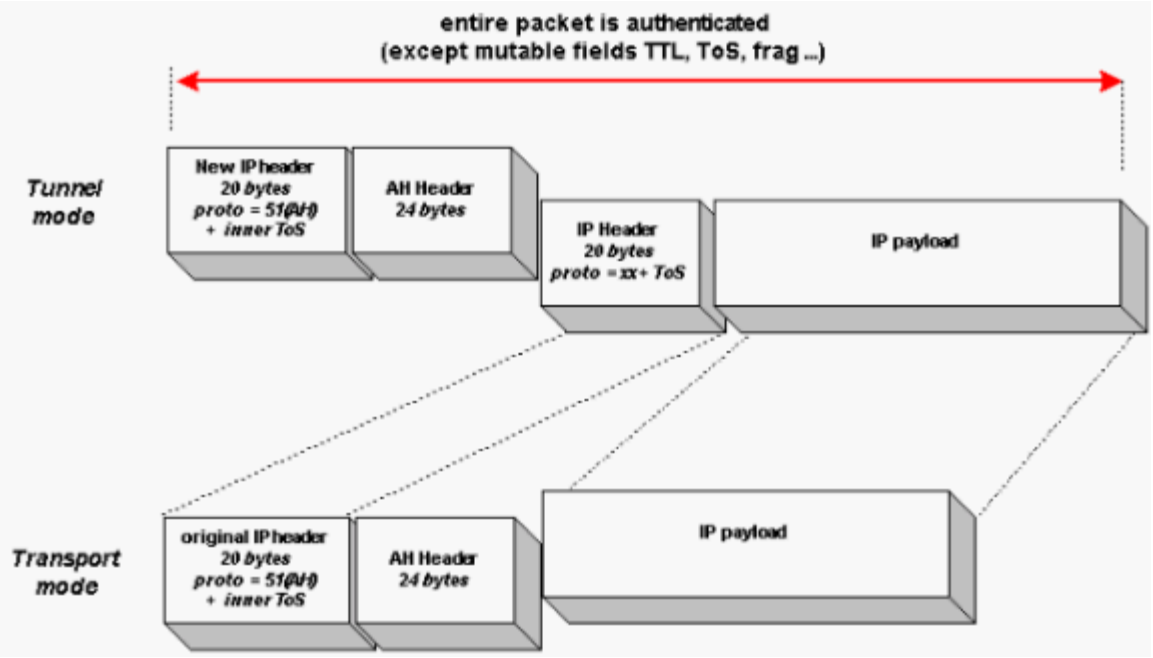
```
P5R0#show crypto ipsec sa address
dest address: 10.1.1.1

!--- Address of the IPSec peer.

   protocol: AH
      spi: 0x93B90183(2478375299)
        transform: ah-md5-hmac ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2808, flow_id: 405, crypto map: testibm
        sa timing: remaining key lifetime (k/sec): (4607901/1654)
        replay detection support: Y
      spi: 0x21A85375(564679541)
        transform: ah-md5-hmac ,

!--- AH  uses the MD5-HMAC algorithm.

        in use settings ={Transport, }
        slot: 0, conn id: 2812, flow_id: 407, crypto map: testibm
        sa timing: remaining key lifetime (k/sec): (4607915/1604)
        replay detection support: Y

   protocol: ESP
      spi: 0xDFF0FEC3(3757113027)
        transform: esp-des ,
        in use settings ={Tunnel, }
        slot: 0, conn id: 2810, flow_id: 405, crypto map: testibm
        sa timing: remaining key lifetime (k/sec): (4607901/1654)
        IV size: 8 bytes
        replay detection support: Y
      spi: 0xDB00B862(3674257506)
        transform: esp-des ,

!--- ESP  uses DES.

        in use settings ={Transport, }

!--- Transport mode accepted for this flow.

        slot: 0, conn id: 2814, flow_id: 407, crypto map: testibm
        sa timing: remaining key lifetime (k/sec): (4607914/1568)
        IV size: 8 bytes
        replay detection support: Y
```

AH and ESP

As noted in the table above, AH and ESP can be used independently or together. However, for most applications only one is sufficient. For both of these protocols, IPSec does not define the specific security algorithms to use. Instead, it provides an open framework to implement industry-standard algorithms.

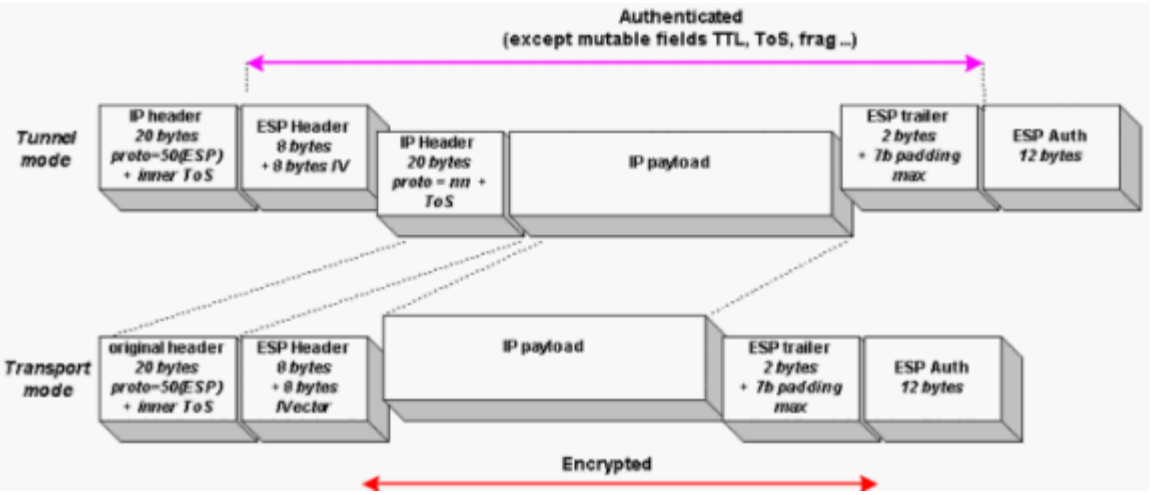
AH provides strong integrity and authentication for IP datagrams using SHA or MD5 hash algorithms. It also provides non-repudiation. The AH is at least 12 bytes and is illustrated in this graphic. The Internet Assigned Numbers Authority (IANA) has assigned protocol number 51 to AH. Thus, in the presence of an AH header with both tunnel mode and transport mode, the IP header uses a value of 51 in the protocol field.

This graphic shows the format of a packet with an IPSec AH. With tunnel mode, the type of service (ToS) byte value is copied automatically from the original IP header to the tunnel header. See the Classify Packets section of this document for more information.



ESP consists of an unencrypted header followed by encrypted data and an encrypted trailer. ESP provides both encryption and authentication. As with AH, ESP supports SHA and MD5 hash algorithms for authentication. It supports DES and 3DES as encryption protocols. The ESP header is at least 8 bytes and is illustrated in this graphic. The IANA has assigned protocol number 50 to ESP. Thus, in the presence of (only) an ESP header with both tunnel mode and transport mode, the IP header uses a value of 50 in the protocol field.

This graphic shows the format of a packet with an IPSec ESP header and trailer. With tunnel mode, the ToS byte value is copied automatically from the original IP header to the tunnel header. See the Classify Packets section of this document for more information.

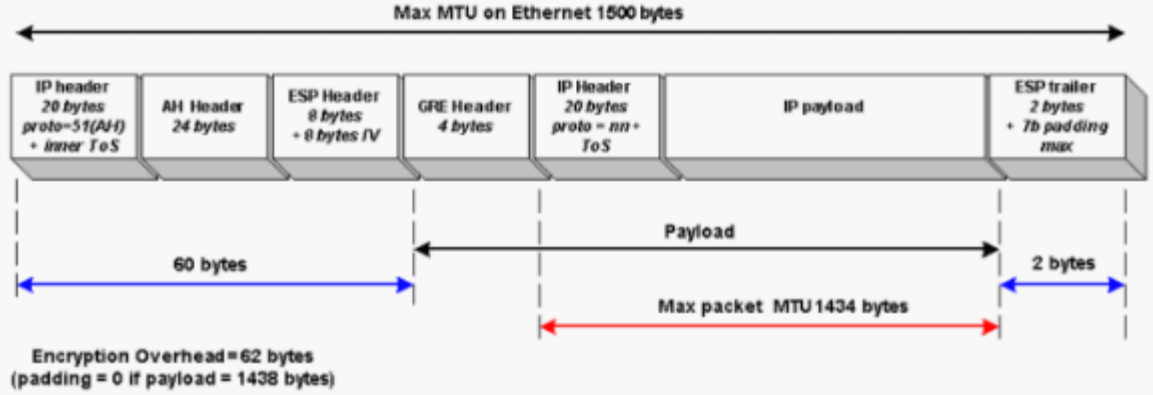


Use GRE Tunnels with IPSec

IPSec does not support multicast or non-IP traffic, such as Internetwork Packet Exchange (IPA) and AppleTalk. The fact that IPSec does not support multicast means that it cannot carry routing protocol information which is multicast like EIGRP (224.0.0.10) and OSPF (224.0.0.5 and 224.0.0.6). GRE is used to encapsulate multicast traffic. This can then be encrypted by IPSec so that routing protocol traffic can flow across a VPN. For a sample configuration of IPSec and GRE, refer to Configuring a GRE Tunnel Over IPSec with OSPF.

The GRE tunnel header introduces a second level of encapsulation. If you use only GRE tunnels and no IPSec, refer to

Quality of Service Options on GRE Tunnel Interfaces. This figure illustrates the packet encapsulated with the IPSec, GRE, and original IP headers:



Classify Packets

Classification defines the process of matching one or more fields in a packet's Layer 2, 3, or 4 headers and then placing that packet in a group or class of traffic. With the help of packet classification, you can partition network traffic into multiple priority levels or classes of service.

When you configure IPSec with GRE, the simplest classification approach is to match on IP precedence or Differentiated Services Code Point (DSCP) values. Cisco IOS Software Release 11.3T introduced support for IPSec and along with it the ToS byte preservation feature. With this feature, the router automatically copies the ToS header value from the original IP packet to the encapsulating IP header when the IPSec in tunnel mode is used.

Cisco PIX Firewall version 5.1 and later and VPN 3000 Series Concentrator version 3.5 and later support ToS byte copying. Section 5.1.2, "Header Construction for Tunnel Mode," in RFC 2401 mandates copying the IP ToS bits.

ToS byte preservation also applies to AH. Also note that ESP in transport mode retains the original IP header, and the original ToS value is transmitted even without ToS byte preservation.

If packets arrive at the router without a set IP precedence or DSCP values, you can mark based on class to remark the packet headers before encryption or encapsulation. When the packets reach the egress interface, the QoS output policy can then match and act on the remarked values.

When you configure a QoS policy based on IP precedence, two policies are applied.

Type of Policy	Actions of Policy	Location of Policy
Input	Mark IP precedence in the original IP header.	Ingress interface
Output	Provide minimum bandwidth guarantees to classes of traffic differentiated by IP precedence.	Egress interface

This table shows a configuration for a QoS policy based on IP precedence:

ToS-Based QoS

Input Policy

```
access-list 150 permit tcp any any eq www
access-list 150 permit tcp any eq www any
access-list 151 permit tcp any any eq telnet
access-list 151 permit tcp any eq telnet any
!
class-map match-any ingress-web
  match access-group 150
class-map match-any ingress-telnet
  match access-group 151
!
policy-map setToS
  class ingress-web
   set ip precedence 1
  class ingress-telnet
   set ip precedence 2
!
interface ethernet 0/0
  service-policy in setToS
```

Output Policy

```
class-map match-any egress-web
  match ip precedence 1
class-map match-any egress-telnet
  match ip precedence 2
!
policy-map useToS
  class egress-web
    bandwidth percent 25
  class egress-telnet
    bandwidth percent 15
!
interface serial 1/0
  bandwidth 512
  service-policy out useToS
  crypto-map TEST
```

Although not shown, you can also apply Weighted Random Early Detection (WRED) to each class as an alternative drop mechanism to tail drop.

The remarked IP precedence value is carried through the network. Thus, make sure that you implement consistent policies through your QoS domain to avoid unexpected classification and performance.

Alternately, you may want to classify traffic based on values other than IP precedence or DSCP. For example, you may want to classify packets based on IP flow or Layer 3 information, such as source and destination IP address. To do this, you must use the QoS for VPNs feature. This feature is enabled with the **qos pre-classify** command and is available for Cisco 7100 series VPN routers and Cisco 7200 series routers since Cisco IOS Software Release 12.1(5)T and for 2600 and 3600 series routers since Cisco IOS Software Release 12.2(2)T. Refer to Configuring QoS for Virtual Private Networks.

The **qos pre-classify** mechanism allows Cisco routers to make a copy of the inner IP header and to run a QoS classification before encryption based on fields in the inner IP header. Without this feature, the classification engine sees only a single encrypted and tunneled flow since all packets that traverse across the same tunnel have the same tunnel header and receive the same treatment in the event of congestion.

If your classification policy matches with the ToS byte, you do not need to use the **qos pre-classify** command since the ToS value is copied to the outer header by default. You can create a simple QoS policy which sorts traffic into classes based on IP precedence. However, to differentiate traffic within a class and to separate it into multiple flow-based queues, the **qos pre-classify** command is required.

Note: ToS byte copying is done by the tunneling mechanism and not by the **qos pre-classify** command.

The **qos pre-classify** command can be applied at various points in your configuration, as illustrated here.

- GRE only - Configure the **qos pre-classify** command on the tunnel interface.

```
interface Tunnel0
  ip address 1.1.1.1 255.255.255.252
  qos pre-classify
  tunnel source 12.2.2.8
  tunnel destination 12.2.2.6
!
interface serial 0/0
  ip address 12.2.2.8 255.255.255.0
  fair-queue
```

- IPSec only - Configure the **qos pre-classify** command under the crypto map.

```
crypto map TEST 10 ipsec-isakmp
  set peer 5.5.5.5
  set transform-set SET
  match address Test
  qos pre-classify
!
interface serial 0/0
  ip address 5.5.5.4 255.255.255.0
  crypto map TEST
  random-detect
  random-detect flow
```

- IPSec and GRE - Configure the **qos pre-classify** command on the tunnel interface and under the crypto map.

```
crypto map TEST 10 ipsec-isakmp
  set peer 12.2.2.6
  set transform-set SET
  match address Test
  qos pre-classify
!
interface Tunnel0
  ip address 1.1.1.1 255.255.255.252
  qos pre-classify
  tunnel source 12.2.2.8
  tunnel destination 12.2.2.6
  crypto map TEST
!
interface serial 0/0
  ip address 12.2.2.8 255.255.255.0
  service-policy out matchPORTnumbers
  crypto map TEST
```

Complete these steps to configure QoS preclassification with IPSec and GRE.

1. Configure a crypto map and specify the **qos pre-classify** command in map configuration mode.

```
crypto map cryptomap_gre1 10 ipsec-isakmp
        set peer 172.32.241.9
        set transform-set transf_GRE1_transport
        match address 130
        qos pre-classify
```

2. Use the **show crypto map** command to confirm your configuration.

```
2621vpn1#show crypto map
Crypto Map: "cryptomap_gre1" idb: Loopback0 local address: 172.31.247.1
Crypto Map "cryptomap_gre1" 10 ipsec-isakmp
```

```
Description: Crypto map on GRE1 tunnel mode transport - 10.240.252.0->3/30
Peer = 172.32.241.9
Extended IP access list 130
    access-list 130 permit gre host 172.31.247.1 host 172.32.241.9
Current peer: 172.32.241.9
Security association lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform sets={ transf_GRE1_transport, }
QOS pre-classification
```

3. Define a GRE tunnel interface and apply the **crypto map** and **qos pre-classify** commands.

```
interface Tunnel0
ip address 10.240.252.1 255.255.255.252
qos pre-classify
tunnel source Loopback0
tunnel destination 172.32.241.9
crypto map cryptomap_gre1
```

4. Use the **show interface tunnel 0** command to confirm that QoS preclassification is enabled.

```
2621vpn1#show interface tunnel 0
Tunnel0 is up, line protocol is up
  Hardware is Tunnel
  Description: VPN resilience test - 1st GRE tunnel Interface mode transport -
10.240.252.0->3/3
  Internet address is 10.240.252.1/30
  Tunnel source 172.31.247.1 (Loopback0), destination 172.32.241.9
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Checksumming of packets disabled,  fast tunneling enabled
  Last input 00:00:04, output 00:00:04, output hang never
  Last clearing of "show interface" counters 00:00:51
  Queueing strategy: fifo (QOS pre-classification)
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
```

The above output illustrates that the tunnel interface continues to use first in, first out (FIFO) as the queuing strategy even with QoS preclassification and fancy queuing enabled. This is illustrated in the **show** command output with the line `Queueing strategy: fifo (QOS pre-classification)`. Both GRE and IPSec tunnels require FIFO queuing since a destination device drops IPSec packets that arrive out of order.

In a VPN environment, you can apply a QoS service policy to the tunnel interface or to the underlying physical interface. The decision of whether you need to configure the **qos pre-classify** command depends on which header and which header values you want to use for classification.

- If you want to classify packets based on the inner header, apply the policy to the tunnel interface without the **qos pre-classify** command.
- If you want to classify packets based on the outer header, apply the policy to the physical interface without the **qos pre-classify** command.
- If you want to classify packets based on the inner header and apply the policy to the physical interface since the physical interface may be a congestion point, apply the policy to a physical interface and enable the **qos pre-classify** command.

As of May 2002, Cisco recommends the application of a hierarchical policy to the physical interface in a VPN environment. Refer to Traffic Policy as a QoS Policy (Hierarchical Traffic Policies) Example. In this configuration, the parent policy includes one class per tunnel to classify or match on all traffic which belongs to that tunnel and uses the **shape** command to limit the output of the tunnel. Classification of flows inside the tunnel is applied with the help of a child policy and can be based on the inner IP header with the **qos pre-classify** command or on the outer IP header without the **qos pre-classify** command. Ensure that the specified bandwidth value in the **shape** command of each of the parent policies for the tunnels does not oversubscribe the interface rate.

Here are the steps to this configuration.

1. Create the classes of the child policy. For example, choose to match on particular DSCP or IP precedence values if those values already are set on the traffic flows.

```
class-map ef
  match ip dscp ef
class-map af
  match ip dscp af31
```

2. Create the child policy map. Specify the classes that you used in step 1. Apply QoS actions to those classes. Examples of such actions include the specification of priority queuing with the **priority** command or the specification of a minimum bandwidth guarantee with the **bandwidth** command.

```
policy-map QoS
  class ef
    priority percent a%
  class af
    bandwidth percent b%
```

3. Create the classes of the parent policy. Use the **access-list** command to specify ACLs that classify all traffic of a particular tunnel by matching on the tunnel source and destination IP addresses. Then create a class map and use the **match access-group** command to reference the ACL.

```
class-map class-tunnel1
  match access-group 101
class-map class-tunnel2
  match access-group 102
class-map class-tunnel3
  match access-group 103
```

4. Create the parent policy map. Specify the classes that you used in step 3. Apply the **shape** command to each class to limit the output of all traffic that comes from the tunnel interface.

```
policy-map main
  class class-tunnel1
    shape average x1 bps
    service-policy QoS
  class class-tunnel2
    shape average x2 bps
    service-policy QoS
  class class-tunnel3
    shape average x3 bps
    service-policy QoS
  class class-default
    shape average x4 bps
```

5. Apply the parent policy map to the physical interface with the help of the **service-policy** command.

```
interface serial 1/0
  service-policy out main
```

Note: If your application requires QoS preclassification in an IPSec or IPSec with GRE environment, enable the **qos pre-classify** command on the crypto map. The match criteria of the parent class should be the same access group that is used by the crypto map. In the example, the match criteria for class-tunnel1 uses the same access group as the crypto map, which you attach to the physical interface or GRE tunnel interface. Cisco supports both software-based encryption and hardware-based encryption, also known as encryption accelerators. This table lists Cisco's crypto hardware accelerators and support for preclassification:

Platform	Encryption Hardware	QoS Preclassify Support
Cisco		Cisco IOS Software Release 12.2T

1700 Series	Wire-speed VPN through hardware encryption.	supports low latency queuing (LLQ) before crypto.
Cisco 2600 and 3600 Series	The Cisco 2600 series supports one internal advanced integration module (AIM) slot. The Cisco 3660 supports two AIM slots. • AIM-VPN/BP (Base Performance) • AIM-VPN/EP (Enhanced Performance) • AIM-VPN/MP (Mid Performance) • AIM-VPN/High Performance (HP) Cisco 7200 Series Router Premier Customer Premise Equipment Application Modular Multiservice Router Virtual Private Network Module for the Cisco 2600 and 3600 Series Cisco 2600 and 3600 VPN Router Bundles	Available, as of Cisco IOS Software Release 12.2(2)T.
Cisco 7100 and 7200 Series	SA-VAM is the VPN Acceleration Module. It installs in a port adapter slot on the Cisco 7200 or 7100 series and installs in the service module slot on the Cisco 7100 series.	Cisco IOS Software Release 12.2T supports the LLQ before crypto feature.
Cisco 7100 and 7200 Series	SA-ISA(=) and SA-ISM(=) are the Integrated Service Adapter and the Integrated Service Module, respectively.	Available in Cisco IOS Software Release 12.2, 12.2T, 12.1E, and 12.0(5)XE.

The behavior of QoS policies changes in the presence of hardware encryption when you modify the switching path of packets inside the router. With hardware encryption, the CPU redirects a packet to the encryption module before it is queued on the outbound interface. Thus, in an IPSec with GRE environment with hardware encryption, there are two

potential congestion points:

- **Crypto queue**—Supports FIFO only. When you use either hardware or software crypto, delay-sensitive packets, such as Voice over IP (VoIP) Real-time Transport Protocol (RTP) streams, you may encounter some latency in the single FIFO queue of the encapsulation process. This latency increases as the delay-sensitive packets arrive when the queue already holds a large amount of data traffic. To minimize any impact, select a Cisco router series with an appropriately scaled architecture and use hardware acceleration. If the crypto engine is not congested, then it poses no problem to have a FIFO for the crypto engine, unless the FIFO is too small to absorb traffic burst. If you run VoIP through the crypto engine, you may want to understand the latency through the engine.

- **Interface-level queue**—Supports fancy-queuing methods. By default, a tunnel interface is a logical interface with a bandwidth parameter of 9 kbps. This bandwidth parameter is used only by upper-layer protocols such as EIGRP and OSPF. It does not actually limit the output rate or interface bandwidth that the tunneled traffic can use. Thus, you may need to implement class-based shaping on the tunnel interface to create "artificial" congestion queues or shaping queues. Class-based shaping limits the output rate and leads to a congested state on the logical tunnel interface. The tunnel interface then begins to queue the excess packets that are held by the shaper, and your fancy queuing policy applies to excess packets.

The hardware crypto engine supports FIFO queuing only. Thus, if you apply a service policy with LLQ on the egress physical interface through which the tunnel traffic is transmitted, ensure that the performance of IPSec processing is greater than the output interface. This allows the interface's priority queuing mechanism to be operative and avoid turning the crypto engine into a FIFO bottleneck.

In Cisco IOS Software Release 12.1, common classification was introduced to ensure that packets match to a single class in a policy-map (refer to Quality of Service Order of Operation for more information). In a VPN environment, one result of this enhancement is that classification of encrypted traffic flows within an IPSec tunnel fails, even when a service policy specifies matching on IP precedence or DSCP values in the tunnel header. This issue is resolved in Cisco IOS Software Release 12.2(10) in Cisco bug IDs CSCdw90486 (registered customers only) and CSCdx08427 (registered customers only) . In Cisco IOS Software releases that include the changes implemented because of these bug, the behavior of classification with hardware and software encryption is now consistent.

This table describes the behavior of MQC features with and without QoS preclassification after the fix. For platforms that do not support preclassification or do not have preclassification enabled, the MQC behavior in the "no preclassify" column is expected. The same definitions used throughout this document for inner and outer header are again used in this table.

	Preclassify	**No Preclassify**
Hardware Crypto Accelerators and CEF		
common classification	inside	outside
set and **police** commands	inside	outside
queuing	inside	outside

flow-based WFQ	inside	outside
Hardware Crypto Accelerators and Process Switching		
common classification	inside	outside
set and **police** commands	inside (1)	outside
queuing	inside	outside
flow-based WFQ	inside	outside
Software Encryption and CEF		
common classification	inside	outside
set and **police** commands	inside	outside
queuing	inside	outside
flow-based WFQ	inside	outside
Software Encryption and Process Switching		
common classification	inside	outside
set and **police** commands	inside (1)	outside
queuing	inside	outside
flow-based WFQ	inside	outside

Note: Although the **set** and **police** commands act on a preclassified class, the marking of packet occurs only at the

outer header.

Sample Configuration

This sample configuration illustrates the commands to create QoS service policies for an IPSec with GRE environment.

- A GRE tunnel connects the IPSec peers, and the tunneled packets are encrypted with the help of IPSec.
- Create an input policy that applies class-based marking to set IP precedence.
- Create an output policy that limits each encrypted tunnel to a maximum bandwidth using class-based shaping and also applies minimum bandwidth guarantees via Class-Based Weighted Fair Queuing (CBWFQ).

Input Policy

Complete these steps to create an input policy that applies class-based marking.

1. Enable CEF with the **ip cef** command. CEF is required for class-based marking. Refer to When Is CEF Required for Quality of Service? for more information.
2. Define criteria on which to sort traffic into classes. This configuration defines class "flow-hi" for Telnet traffic and class "flow-low" for ICMP traffic.
3. Define a policy map and define QoS actions to the defined classes.
4. Apply your policy to the interface. In this case, it is the serial subinterface.

This table shows a configuration for a class-based marking input policy.

Class-Based Marking Input Policy

```
ip cef

!
class-map match-any flow-low
 match protocol icmp
!
class-map match-any flow-hi
  match protocol telnet

!
policy-map qos-in
 class flow-hi
  set ip precedence 4
!
class flow-low
  set ip precedence 2
!
int s0/0.1
 service-policy input qos-in


router#show policy-map interface s0/0.1

Serial0/0.1

Service-policy input: qos-in)

!--- Apply input policy named "qos-in."
```

```
     Class-map: flow-hi (match-any)
      447 packets, 227851 bytes
      30 second offered rate 5000 bps, drop rate 0 bps

!--- Input rate for class named "flow-hi."


      Match: protocol telnet
        447 packets, 227851 bytes
        30 second rate 5000 bps

!--- Input rate for class named "flow-hi."

      QoS Set
        ip precedence 4
          Packets marked 447

!--- Number of packets marked.


   Class-map: flow-low (match-any
      237 packets, 337898 bytes
      30 second offered rate 21000 bps, drop rate 0 bps
      Match: protocol icmp
        237 packets, 337898 bytes
        30 second rate 21000 bps
      QoS Set
        ip precedence 2
          Packets marked 237

    Class-map: class-default (match-any)

!--- The default class is automatically defined.

      1 packets, 48 bytes
      30 second offered rate 0 bps, drop rate 0 bps
      Match: any
```

Output Policy

This configuration creates a hierarchical output policy, also known as a nested policy (refer to Traffic Policy as a QoS Policy (Hierarchical Traffic Policies) Example for more information). The "parent" policy applies class-based shaping to limit the overall output rate of the tunnel interface, and the "child" policy applies minimum bandwidth guarantees using CBWFQ to the queued excess.

These are the steps for this configuration.

1. Create the child policy.
    - This configuration defines class "ipsec-hi" for the remarked Telnet traffic and class "ipsec-low" for the remarked ICMP traffic.
    - This configuration uses policy map "ipsec-flow" to apply CBWFQ to the Telnet and ICMP traffic using the **bandwidth** command.
2. Create the parent policy. This configuration defines class "ipsec" and shapes that match traffic to 16 kbps.
3. Apply the child policy within the parent policy. This configuration applies the child policy "ipsec-flow" as a QoS action within the parent policy "qos-out." The QoS action is CBWFQ to packets retained and queued by the shaper.
4. Apply the parent policy to the interface. In this case, it is the serial subinterface.

This table shows a configuration for an output policy to shape and apply CBWFQ, based on the remarked values.

Shaping and CBWFQ Outbound Policy

```
class-map match-all ipsec-hi
  match ip precedence 4
class-map match-all ipsec-low
  match ip precedence 2
!
policy-map ipsec-flow
  class ipsec-hi
   bandwidth 8
  class ipsec-low
   bandwidth 8
!
class-map match-all ipsec
 match protocol gre
!
policy-map qos-out
 class ipsec
  shape average 16000
   service-policy ipsec-flow
!
int fa0/0
 service-policy output qos-out

!--- Apply the policy to the physical interface through


!--- which the tunnel traffic is transmitted.

router#show policy-map interface fast 0/0
 FastEthernet0/0

  Service-policy output: qos-out

!--- "Parent" policy named "qos-out."


    Class-map: ipsec (match-all)
      1422 packets, 1390125 bytes
      30 second offered rate 38000 bps, drop rate 0 bps

!--- Egress rate before shaping.


      Match: protocol gre
      Traffic Shaping

      Target  Byte   Sustain   Excess    Interval
Increment Adapt
      Rate    Limit  bits/int  bits/int  (ms)
(bytes)   Active
      16000   2000   8000      8000      500       1000
-


      Queue  Packets   Bytes   Packets  Bytes
Shaping
      Depth                     Delayed  Delayed
Active
        69     641      611106  582      535364    yes

  Service-policy : ipsec-flow

!--- "Child" policy named "ipsec-flow."


        Class-map: ipsec-hi (match-all)
          788 packets, 464485 bytes
          30 second offered rate 15000 bps, drop rate 0
bps
          Match: ip precedence 4
          Weighted Fair Queueing
            Output Queue: Conversation 25
            Bandwidth 8 (kbps) Max Threshold 64
(packets)
            (pkts matched/bytes matched) 389/241922
            (depth/total drops/no-buffer drops) 4/0/0
```

```
        Class-map: ipsec-low (match-all)
          634 packets, 925640 bytes
          30 second offered rate 25000 bps, drop rate 0
bps
          Match: ip precedence 2
          Weighted Fair Queueing
            Output Queue: Conversation 26
            Bandwidth 8 (kbps) Max Threshold 64
(packets)
            (pkts matched/bytes matched) 270/400140
            (depth/total drops/no-buffer drops) 64/2/0

        Class-map: class-default (match-any)
          0 packets, 0 bytes
          30 second offered rate 0 bps, drop rate 0 bps
          Match: any

       Class-map: class-default (match-any)
         115 packets, 14827 bytes
         30 second offered rate 0 bps, drop rate 0 bps
         Match: any
```

Note: The parent policy uses the **match protocol gre** command to specify matching on the protocol value assigned to GRE in the IP header. Based on the order of execution of Cisco IOS features, ACLs and QoS features see unencrypted packets. Thus, the configuration of ACLs that match on the AH or ESP protocol value (51 and 50, respectively) do not work (Cisco bug IDs CSCdu63385 (registered customers only) and CSCdv20737 (registered customers only)). This restriction applies to both hardware and software encryption. In rare cases, QoS features classified packets based on the modified IP header of encrypted packets on a router configured for CEF switching. The reason is that CEF packets actually were being process switched when the crypto code and CEF wrongly could not locate a valid CEF adjacency.

Note: If you use the **match protocol** command in a class map to match on non-IP protocols, such as IPX and AppleTalk, and also enable **qos pre-classify** to match on values in the inner header, classification based on the inner header do not work.

Restrictions and Related Issues

This section discusses known issues and workarounds related to the application of crypto and QoS on the same router.

QoS and Anti-Replay Protection

Some crypto transform sets provide anti-replay protection, which works when you apply a sequence number to the crypto header. On a congested interface with fancy queuing, a low-priority packet may be delayed in a queue and then arrive at the router that decrypts after the anti-replay window has been exceeded. In this case, the device that receives will drop the packet. In addition, if an encrypted packet arrives at the destination out of sequence by a certain window (currently set to 64 packets), the packet is dropped. Cisco is currently on the lookout for methods to overcome these limitations. Note that anti-replay protection cannot be disabled from these transforms in which it is implemented.

- esp-sha-hmac
- esp-md5-hmac
- ah-md5-hmac
- ah-sha-hmac

Use the **show crypto ipsec sa** command to determine whether anti-replay support is enabled for each IPSec SA.

```
2611-ch5#show crypto ipsec sa
interface: Tunnel0
Crypto map tag: Test, local addr. 12.2.2.6
```

```
inbound esp sas:
spi: 0xDE92271(233382513)
transform: esp-des esp-sha-hmac ,
in use settings ={Transport, }
slot: 0, conn id: 2000, flow_id: 1, crypto map: Test
sa timing: remaining key lifetime (k/sec): (4607996/99)
IV size: 8 bytes
replay detection support: Y
```

NBAR

NBAR is not configurable on logical interfaces where tunneling or encryption is used. It also is not supported on any physical interface configured with a crypto map. Thus, you cannot use NBAR to classify traffic based on higher-layer packet information such as a URL or Web server hostname for any QoS policy where GRE and/or IPSec are being used. This restriction results from the number of bytes of the packet header that the pre-classify feature saves and then refers. Specifically, QoS preclassification calls an API in IOS before a packet is encapsulated. This API takes a copy of the original packet header information. When the packet eventually hits the egress QoS function, QoS can be applied to the packet based on any of the saved information such as TCP port or real destination IP address.

Double Accounting

The classification counters in the output of the **show policy-map interface** command may display double the number of known tunneled packets that are encrypted in a configuration with CEF, GRE, and IPSec. This output illustrates this condition.

```
router#show policy-map interface fa0/0
  FastEthernet0/0

   Service-policy output: qos-out

     Class-map: ipsec (match-all)
       44 packets, 8580 bytes
       30 second offered rate 1000 bps, drop rate 0 bps
       Match: protocol gre
       Traffic Shaping
         Target    Byte   Sustain   Excess    Interval  Increment Adapt
         Rate      Limit  bits/int  bits/int  (ms)      (bytes)   Active
         16000     2000   8000      8000      500       1000      -

         Queue     Packets   Bytes     Packets   Bytes     Shaping
         Depth                         Delayed   Delayed   Active
         0         22        4796      0         0         no
```

This condition results from output packet classification that happens once in the CEF switching path and again when the crypto process dequeues the packet. This problem is resolved in these Cisco bug IDs.

- CSCdu17976 (registered customers only) - Resolves this problem by adding a flag which marks the packet as having already been classified.
- CSCdv79109 (registered customers only) - CEF switching and Cisco IOS Software Release 12.2 mainline.
- CSCdt62225 (registered customers only) - Fast switching and Cisco IOS Software Release 12.2 mainline.

In addition, when both GRE tunnels and IPSec are configured, a packet is run two times through the CEF lookup process, once after GRE encapsulation and once after IPSec encapsulation. Since the packet is transmitted only after IPSec encapsulation, CEF per-packet load balancing fails, and packets always use the same interface.

Software Encryption and Fast Switching/CEF

When you use software encryption, the QoS preclassify feature, and CBWFQ, fast-switched packets may not be classified properly. This problem is seen in the output of the **show policy-map interface** command:

```
3640-ch1#show policy-map interface
 {snip}
      Class-map: precedence (match-all)
        5 packets, 520 bytes

!--- Five packets matched the class.

        30 second offered rate 0 bps, drop rate 25000 bps
        Match: ip precedence 5
        Weighted Fair Queueing
          Output Queue: Conversation 26
          Bandwidth 10 (%)
          Bandwidth 1 (kbps) Max Threshold 64 (packets)
          (pkts matched/bytes matched) 11756/14350192

!--- Many packets actually are  queued.

          (depth/total drops/no-buffer drops) 63/6359/0
```

As a workaround, use a hardware encryption module, force process switching with the **no ip route-cache cef** command, or disable QoS preclassify. This results in the fair queuing system seeing a single encrypted flow in the class-default class.

This issue is resolved in Cisco bug ID CSCdw28771 (registered customers only) .

Legacy Priority Queuing and QoS PreClassify

Cisco's legacy priority queuing feature, which uses the **priority-list** and **priority-group** commands, and the QoS preclassify feature are not supported together. Instead, implement LLQ by configuring a policy map with the **priority** command of the MQC.

Hardware Encryption and QoS

These are resolved issues with hardware encryption and QoS.

- Cisco bug ID CSCdv25358 (registered customers only) - When you use the **rate-limit** command to implement traffic policing through the Cisco legacy committed access rate (CAR) feature, the **qos pre-classify** command does not work when hardware encryption also is used. This feature combination prevents encryption from happening. As a workaround, implement class-based policing with the help of the **police** command in a policy-map configured with the modular QoS command line interface (CLI) (MQC). Other QoS features (MQC or non-MQC) are not affected.

- Cisco bug ID CSCdw29595 (registered customers only) - The performance of the encryption path degrades when Cisco IOS Software Release 12.2(6.8) is used with a hardware encryption card. The loss in performance occurs because encrypted packets are process-switched instead of being fast-switched. This condition occurs when IPSec is applied to the interfaces while the hardware encryption card is used. There is no workaround.

- Cisco bug ID CSCdw30566 (registered customers only) - In an IPSec with GRE environment, if you enable CEF, it leads to reduced forwarding performance since packets actually are process-switched. This condition results from how CEF processed packets after they were GRE-encapsulated. As a workaround, disable CEF, and allow the packets to be fast-switched.

When you use crypto accelerators, see the Classify Packets section of this document and the discussion of changes implemented in Cisco bug IDs CSCdw90486 (registered customers only) and CSCdx08427 (registered customers only) .

Cisco Support Community - Featured Conversations

Cisco Support Community is a forum for you to ask and answer questions, share suggestions, and collaborate with your peers. Below are just some of the most recent and relevant conversations happening right now.

Related Information

- **QoS Support**
- **RFC 2401 - Security Architecture for the Internet Protocol**
- **RFC 2402 - Authentication Header**
- **RFC 2406 - IP Encapsulating Security Payload (ESP)**
- **Configuring and Troubleshooting Cisco Network-Layer Encryption: Background - Part 1**
- **Quality of Service Options on GRE Tunnel Interfaces**
- **Configuring a GRE Tunnel over IPsec with OSPF**
- **Technical Support & Documentation - Cisco Systems**