GROUP ENCRYPTED TRANSPORT VPN (GET VPN)

DESIGN AND IMPLEMENTATION GUIDE

Document Version 1.0
August 2008

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Abstract

This Design and Implementation Guide (DIG) describes the design and implementation of Group Encryption Transport VPN (GET VPN) networks in enterprise environments. The guide covers scalability and performance considerations, and also discusses design considerations related to high availability, IPsec, the placement and configuration of GET VPN components, and resiliency. This guide provides an overview of GET VPN and discusses key aspects of the technology, but this guide does not describe GET VPN technology in depth. For more details, visit http://www.cisco.com/go/getvpn.

Key Technologies

GET VPN

Contents

1 About Group Encrypted Transport Virtual Private Networks

Networks have become critical strategic assets and lifelines for running successful enterprises. Today's networks not only support critical applications, but also support voice and video infrastructures. Applications and technologies, such as distributed computing and voice and video over IP, now require instantaneous branch-to-branch communication. Because of these requirements, the traditional hub-and-spoke topology of enterprise networks is no longer sufficient. Enterprises must implement the any-to-any connectivity model provided by IP virtual private networks (VPNs) and virtual private LAN services (VPLS) networks.

Although IP VPN and VPLS services built with Multiprotocol Label Switching (MPLS) separate enterprise traffic from the public Internet to provide some security, in recent years government regulations, such as Health Insurance Portability and Accountability Act (HIPAA), Gramm-Leach-Bliley Act (GLBA), and Payment Card Industry Data Security Standard (PCI DSS), mandate encryption even over private IP networks.

Cisco IOS offers several IP security (IPsec) tunnel-based encryption solutions (for example, Site to Site IPsec, IPsec/generic routing encapsulation (GRE), and Dynamic Multipoint VPN (DMVPN) that can be deployed over an MPLS VPN, VPLS or shared IP networks. Traditional tunnel-based encryption solutions are point-to-point.

To provide a true full mesh or even dense partial mesh of connectivity, tunnel-based solutions require the provisioning of a complex connectivity mesh. Such a complex mesh not only has higher processor and memory requirements, but is difficult to provision, troubleshoot, and manage. Some provisioning overhead can be reduced using DMVPN. However, DMVPN requires overlaying a secondary routing infrastructure through the tunnels, which results in suboptimal routing while the dynamic tunnels are built. The overlay routing topology also reduces the inherent scalability of the underlying IP VPN network topology.

Traditional point-to-point IPsec tunneling solutions suffer from multicast replication issues because multicast replication must be performed before tunnel encapsulation and encryption at the IPsec CE (customer edge) router closest to the multicast source. Multicast replication cannot be performed in the provider network because encapsulated multicasts appear to the core network as unicast data.

Cisco's Group Encrypted Transport VPN (GET VPN) introduces the concept of a *trusted group* to eliminate point-to-point tunnels and their associated overlay routing. All group members (GMs) share a common security association (SA), also known as a group SA. This enables GMs to decrypt traffic that was encrypted by any other GM. (Note that IPsec CE acts as a GM.) In GET VPN networks, there is no need to negotiate point-to- point IPsec tunnels between the members of a group, because GET VPN is "tunnel-less."

The IETF standard RFC-3547 Group Domain of Interpretation (GDOI) is an integral part of GET VPN. The GDOI protocol was introduced in 12.4(2)T but the GET VPN solution with several enhancements was released in 12.4(11)T. See 1.2.1, "GDOI," for more information about GDOI.

1.1 Key GET VPN Benefits

GET VPN provides the following key benefits:

- Instantaneous large-scale any-to-any IP connectivity using a group IPsec security paradigm

- Takes advantage of underlying IP VPN routing infrastructure and does not require an overlay routing control plane
- Seamlessly integrates with multicast infrastructures without the multicast replication issues typically seen in traditional tunnel-based IPsec solutions.
- Preserves the IP source and destination addresses during the IPsec encryption and encapsulation process. Therefore GET VPN integrates very well with features such as QoS and traffic engineering.

1.2 Technology Overview

The GET VPN solution is based on both open standards and Cisco patented innovative technology which helps utilize the power of underlying MPLS/shared IP networks. In addition to leveraging the existing IKE, IPsec and multicast technologies, GET VPN solution relies on following core building blocks to provide the required functionality:

- GDOI (RFC 3547)
- Key servers (KSs)
- Cooperative (COOP) KSs
- GMs
- IP tunnel header preservation
- Group security association
- Rekey mechanism
- Time-based anti-replay (TBAR)

1.2.1 GDOI

The GDOI group key management protocol is used to provide a set of cryptographic keys and policies to a group of devices. In a GET VPN network, GDOI is used to distribute common IPsec keys to a group of enterprise VPN gateways that must communicate securely. These keys are periodically refreshed and are updated on all the VPN gateways using a process called “rekey.”

The GDOI protocol is protected by a Phase 1 Internet Key Exchange (IKE) SA. All participating VPN gateways must authenticate themselves to the device providing keys using IKE. All IKE authentication methods, for example, pre-shared keys (PSKs) and public key infrastructure (PKI), are supported for initial authentication. After the VPN gateways are authenticated and provided with the appropriate security keys via the IKE SA, the IKE SA expires and GDOI is used to update the GMs in a more scalable and efficient manner.

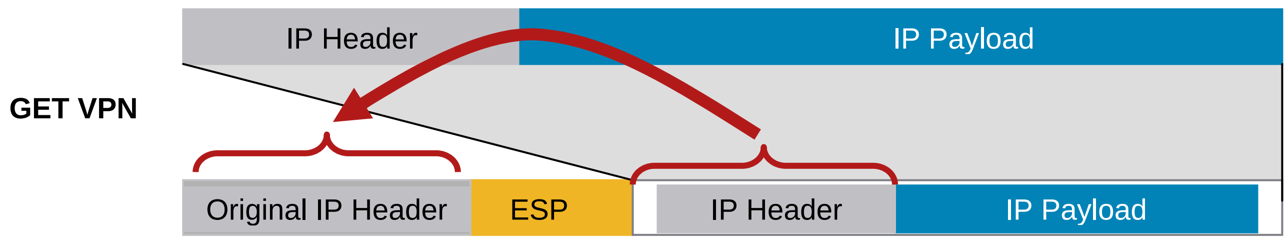
Tip: For more information about GDOI, refer to RFC 3547.

GDOI introduces two different encryption keys. One key secures the GET VPN control plane; the other key secures the data traffic. The key used to secure the control plane is commonly called the Key Encryption Key (KEK), and the key used to encrypt data traffic is known as Traffic Encryption Key (TEK).

1.2.2 Tunnel Header Preservation

In traditional IPsec, tunnel endpoint addresses are used as new packet source and destination. The packet is then routed over the IP infrastructure, using the encrypting gateway source IP address and the decrypting gateway destination IP address. In the case of GET VPN, IPsec protected data packets encapsulate the original source and destination packet addresses of the host in the outer IP header to "preserve" the IP address.

Figure 1-1 Tunnel Header Preservation



The biggest advantage of tunnel header preservation is the ability to route encrypted packets using the underlying network routing infrastructure. The branch high availability (HA) provided by the IP, VPLS, or MPLS VPN infrastructure (dual spokes, dual links, and so on) integrates seamlessly with the GET VPN solution. There is no need to provide HA at the IPsec level (dual hubs, stateful IPsec HA, and so on).

Because tunnel header preservation is combined with group SAs, multicast replication can be offloaded to the provider network. Because every GM shares the same SA, the IPsec router closest to the multicast source does not need to replicate packets to all its peers, and is no longer subject to multicast replication issues seen in traditional IPsec solutions.

It is worth noting that tunnel header preservation seems very similar to IPsec transport mode. However, the underlying IPsec mode of operation is IPsec tunnel mode. While IPsec transport mode reuses the original IP header and therefore adds less overhead to an IP packet (5% for IMIX packets; 1% for 1400-byte packets), IPsec transport mode suffers from fragmentation and reassembly limitations and must not be used in deployments where encrypted or clear packets might require fragmentation.

Note: Because of tunnel header preservation, GET VPN solution is very well suited for MPLS, Layer-2 (L2), or an IP infrastructure with end to end IP connectivity. However, GET VPN is generally not a good candidate for deployment over the Internet because enterprise host IP addresses are typically not routable, and network address translation (NAT) functions interfere with tunnel header preservation.

1.2.3 KSs

A key server (KS) is an IOS device responsible for creating and maintaining the GET VPN control plane. All encryption policies, such as interesting traffic, encryption protocols, security association, rekey timers, and so on, are centrally defined on the KS and are pushed down to all GMs at registration time.

GMs authenticate with the KS using IKE Phase 1 (pre-shared keys or PKI) and then download the encryption policies and keys required for GET VPN operation. The KS is also responsible for refreshing and distributing the keys.

Unlike traditional IPsec, interesting traffic defined on the KS (using an access control list (ACL)) is downloaded to every GM, whether or not the GM owns that network. It is recommended to summarize GM networks into as few entries as possible, and to strive for a symmetric policy. For example, if all LAN addresses on the GMs are within the 10.0.0.0/8 network (10.1.1.0/24, 10.1.2.0/24, and so on), it is better to define interesting traffic as “permit 10.0.0.0/8 to 10.0.0.0/8” as apposed to “permit 10.1.1.0/24 to 10.1.2.0/24”, “10.1.1.0/24 to 10.1.3.0/24,”and so on.

Asymmetric policies lead to a geometric expansion in the number of ACL entries. An aggregate policy that serves the most GMs is ideal. The most complete aggregate policy is **`permit ip any any`** This policy encrypts **all** traffic leaving the GM crypto interface. Therefore, exceptions must be made (that is, deny entries) to exclude encryption of control plane traffic and management plane necessary to bootstrap the GM.

Note: A device acting as a KS can not be configured as a GM

1.2.4 GMs

A GM is an IOS router responsible for actual encryption and decryption i.e. a device responsible to handle GET VPN data plane. A GM is only configured with IKE phase 1 parameters and KS/Group information. As mentioned before, encryption policies are defined centrally on the KS and downloaded to the GM at the time of registration. Based on these downloaded policies, GM decides whether traffic needs to be encrypted or decrypted and what keys to use. In a GET VPN network, GM policies are dictated by the KS but in some instances, a GM can be configured to locally override some of these policies. Any global policy (including both permit and deny entries) defined on the KS affects all the members of the group whether it is applicable to them or not and therefore some policies make more sense when defined locally. As an example, if a handful of GMs in the group are running a different routing protocol, a local entry can be added to these GMs to bypass encryption of the routing protocol traffic instead of defining the policy globally at the KS level.

1.2.5 Group SA

Unlike traditional IPsec encryption solutions, GET VPN uses the concept of group SA. All members in the GET VPN group can communicate with each other using a common encryption policy and a shared SA. With a common encryption policy and a shared SA, there is no need to negotiate IPsec between GMs; this reduces the resource load on the IPsec routers. Traditional GM scalability (number of tunnels and associated SA) does not apply to GET VPN GMs.

Note: In a GET VPN group, up to 100 ACL permit entries can be used to define interesting traffic for encryption. Each permit entry results in a pair of IPsec SAs; the maximum number of IPsec SAs in a group can not exceed 200.

It is a **best practice** to summarize interesting traffic to as few permit entries as possible, and to build symmetric policies. Unlike traditional IPSec policies, where source and destination address ranges must be uniquely defined, GET VPN is optimized when the source and destination address range are the same. This minimizes the number of policy permutations, making GET VPN very efficient.

1.2.6 Rekey Process

As mentioned above, the KS is not only responsible for creating the encryption policies and keys, but also for refreshing keys and distribute them to GMs. The process of sending out new keys when existing keys are about to expire, is known as the rekey process. GET VPN supports two types of rekey messages: unicast and multicast.

If a GM does not receive rekey information from the KS (for example, the KS is down or network connectivity is broken), the GM tries to reregister to an ordered set of KSs 60 seconds before the existing IPsec SAs expire. If reregistration is successful, the GM receives new SAs as part of the reregistration process and traffic in the data plane flows without disruption. If reregistration is unsuccessful (the preferred KS is unavailable), the GM tries three more times, at 10-second intervals, to establish a connection with the KS. If all attempts to contact the preferred KS fail, the GM tries the next KS in the ordered list 20 seconds before existing IPsec SAs expire.

1.2.6.1 Unicast Rekey

In the unicast rekey process, a KS generates a rekey message and sends multiple copies of the message, one copy to each GM. Upon receiving the rekey message, a GM sends an ACK message to the KS. This ACK mechanism not only ensures that the list of active GMs on the KS is current, but also ensures that the rekey message is sent only to active GMs.

A KS can be configured to retransmit a rekey packet to overcome transient defects in the network. If a GM does not acknowledge three consecutive rekeys (retransmissions are considered part of the rekey), the KS removes the GM from its active GM database and stops sending rekey messages to that GM.

Note: Unicast rekey retransmission occur only when a GM does not ACK a rekey.

1.2.6.2 Multicast Rekey

In the multicast rekey process, a KS generates a rekey message and sends one copy of the message to a multicast group address that is predefined in the configuration. Each GM joins the multicast group at registration time, so each GM receives a copy of the rekey message.

Unlike unicast rekey, multicast rekey does not have an ACK mechanism. The KS does not maintain a list of active GMs. Multicast rekey uses the same low CPU overhead whether there is one GM in the group or a few thousand. Just like unicast rekey, KS can be configured to retransmit a multicast rekey packet to overcome transient network defects.

Note: Multicast must be enabled in the core network for multicast rekey to work in the GET VPN control plane.

1.2.7 COOP KSs

The KS is the most important entity in the GET VPN network because the KS maintains the control plane. Therefore, a single KS is a single point of failure for an entire GET VPN network. Because redundancy is an important consideration for KSs, GET VPN supports multiple KSs, called cooperative (COOP) KSs, to ensure seamless fault recovery if a KS fails or becomes unreachable.

A GM can be configured to register to any available KS from a list of all COOP KSs. GM configuration determines the registration order. The KS defined first is contacted first, followed by the second defined KS, and so on.

Note: The COOP protocol is configured on a per GDOI group basis. A KS that is configured with multiple GDOI groups can maintain multiple unique COOP relationships with disparate KSs.

When COOP KSs boot, all KSs assume a “secondary” role and begin an election process. One KS, typically the one having the highest priority, is elected as a “primary” KS. The other KSs remain in the secondary state. The primary KS is responsible for creating and distributing group policies to all GMs, and to periodically synchronize the COOP KSs.

Note: GMs can register to any available KS (primary or secondary), but only the primary KS sends rekey messages. It is a best practice to distribute GM registration to all available COOP KSs to reduce the IKE processing load on a single KS. See 3.7.3.3, “Balancing GM Registrations among COOP KSs,” for details.

Cooperative KSs exchange one-way announcement messages (primary to secondary). If a secondary KS does not hear from the primary KS for a certain length of time, the secondary KS tries to contact the primary KS and requests updated information. If the primary KS does not respond, or if the secondary KS does not hear from the primary KS, a COOP reelection is triggered and a new primary KS is elected.

Up to eight KSs can be defined as COOP KSs, but more than four COOP servers are seldom required. Since rekey information is generated and distributed from a single primary KS, the advantage of deploying more than two KSs is the ability to handle registration load in case of a network failure and reregistration taking place at the same time. This is especially important when using Public Key Infrastructure (PKI) because IKE negotiation using PKI requires a lot more CPU power compared to IKE negotiation using pre-shared keys (PSKs).

Tip: It is a best practice to enable periodic ISAKMP keepalives between the KSs so that primary KS can track and display the state of the other secondary KSs. IKE Keepalives between GM and KS is not required and is not supported.

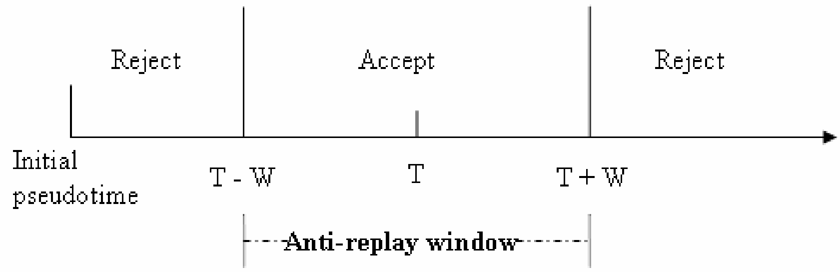
1.2.8 Time Based Anti-Replay

In traditional IPsec solutions, anti-replay capabilities prevent a malicious third party from capturing IPsec packets and relaying those packets at a later time to launch a denial of service attack against the IPsec endpoints. These traditional IPsec solutions use a counter based sliding window protocol: The sender sends a packet with a sequence number, and the receiver uses the sliding window to determine whether a packet is acceptable, or has arrived out-of-sequence and is outside the window of acceptable packets.

Because we use the group SA in GET VPN, counter-based anti-replay is ineffective. A new method to guard against replay-attacks is required. GET VPN uses time-based anti-replay (TBAR), which is based on a pseudo-time clock that is maintained on the KS. An advantage of using pseudotime for TBAR is that there is no need to synchronize time on all the GET VPN devices using NTP.

The primary KS is responsible for establishing and maintaining the pseudo-time for a group. The primary KS must also keep pseudotime synchronized on all GMs via rekey updates. Every GM includes its pseudo-time as a time stamp in the data packets. A receiving VPN gateway then compares time stamp of the received packet with the GM reference pseudotime clock it maintains for the group. If the packet arrived too late, it is dropped.

Figure 1-2 Time Based Anti-Replay

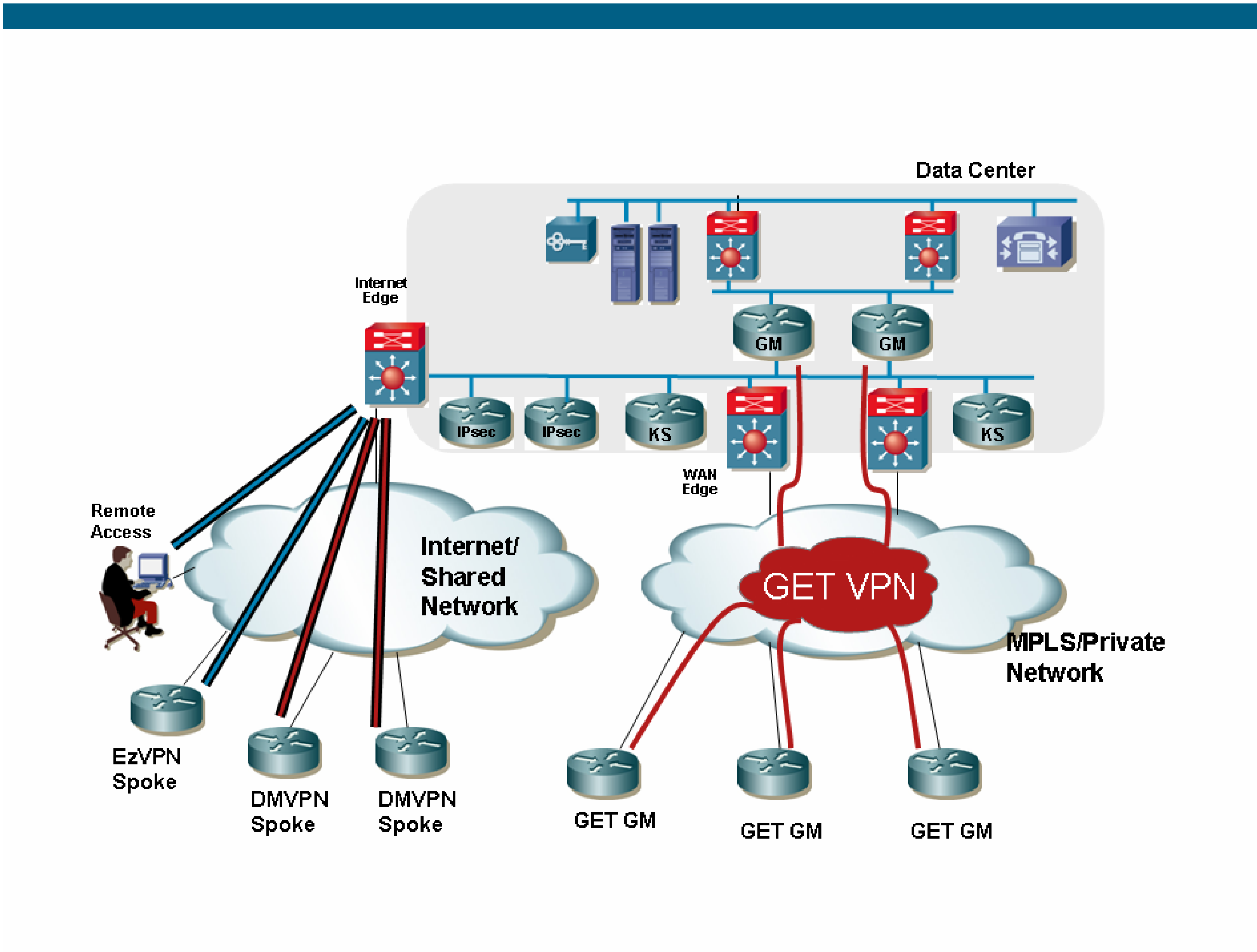


Tip: GET VPN is typically deployed over a private WAN (VPLS, MPLS, and so on). While the threat of anti-replay attack is minimal, it is a best practice to enable TBAR. TBAR is currently (as of 12.4(15)T5) not supported on 7200 NPE-G2/VSA.In networks with VSA, TBAR should be completely disabled. TBAR is enabled groupwide on the KS.

1.3 GET VPN Solution Positioning

Figure 1-3 illustrates the positioning of the GET VPN solution.

Figure 1-3 GET VPN Solution Positioning



As noted, GET VPN is not well suited for deployment over the Internet, so a tunnel-based IPsec solution such as DMVPN, EzVPN, or Site-to-Site VPNs should be deployed over the Internet.

1.4 GET VPN Solution Comparison

Table 1-1 provides a basic comparison of EzVPN, DMVPN and GET VPN technologies. Consult the detailed documentation about these technologies for further information.

Table 1-1 GET VPN Solution Comparison

	EzVPN	DMVPN	GET VPN
Infrastructure Network	Public Internet Transport	Public Internet Transport	Private IP Transport
Network Style	Hub-Spoke; (Client to Site)	Hub-Spoke and Spoke-to-Spoke; (Site-to-Site)	Any-to-Any; (Site-to-Site)
Routing	Reverse-route Injection	Dynamic routing on tunnels	Dynamic routing on IP WAN
Failover Redundancy	Stateful Hub Crypto Failover	Route Distribution Model	Route Distribution Model + Stateful
Encryption Style	Peer-to-Peer Protection	Peer-to-Peer Protection	Group Protection
IP Multicast	Multicast replication at hub	Multicast replication at hub	Multicast replication in IP WAN network

1.5 Further Reading

While this document provides an overview of the GET VPN technology and discusses key aspects of the technology, it should not be considered an in-depth technology primer. For further details on the GET VPN technology, refer to following documentation:

GET VPN Documentation on www.cisco.com

http://www.cisco.com/go/getvpn

GET VPN Feature Documentation

http://www.cisco.com/en/US/docs/ios/12_4t/12_4t11/htgetvpn.html

2 GET VPN Configuration

This chapter describes the basic configuration and verification of the following GET VPN components:

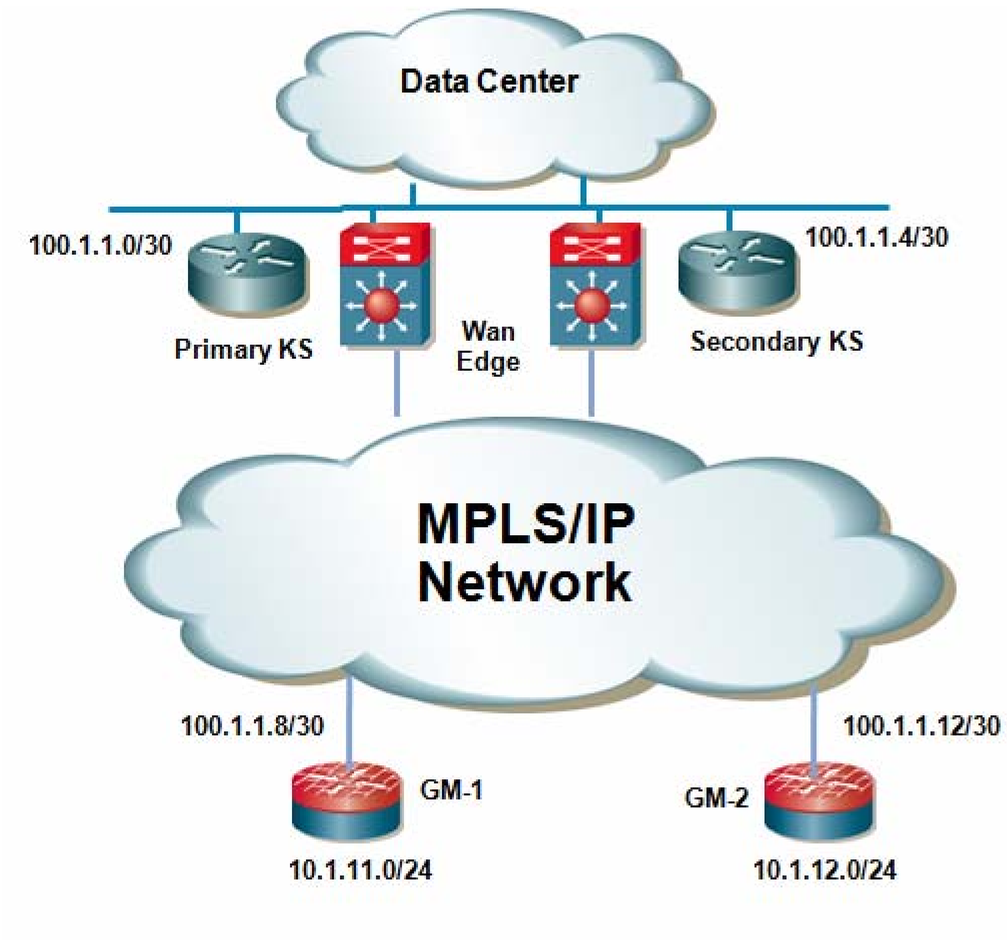
- Key servers (KSs) and group members (GMs)
- Pre-shared key (PSK) and public key infrastructure (PKI)
- Unicast and multicast rekey
- Cooperative (COOP) KSs

Chapter 3, “System Design” and Chapter 4, “Enterprise Deployment,” provide detailed descriptions of GET VPN design and deployment, respectively.

2.1 Implementing GET VPN

Figure 2-1 illustrates a high-level GET VPN system configuration topology.

Figure 2-1 System Configuration Topology



The topology in Figure 2-1 is used to setup the GET VPN network. The IP VPN core interconnects VPN sites as shown in the figure. The CE/CPE routers (GMs 1 and 2) on each VPN site are grouped into a GDOI group. Therefore, all KSs and GMs are part of the same VPN. KS-1 is the primary KS and KS-2 is the secondary KS.

2.2 KS Configuration

It is assumed that all connectivity (default routes, routing protocols (RPs), and so on) are set up before the KS is configured.

2.2.1 Configuring IKE Phase 1

Internet Key Exchange (IKE) Phase 1 comprises the following

- Configuring Internet Security Association and Key Management Protocol (ISAKMP) policy
- Configuring the authentication method

2.2.1.1 Configuring ISAKMP Policy

The IKE Phase 1 (ISAKMP policy) configuration follows:

```
crypto isakmp policy 10
 encr aes
 group 2
```

2.2.1.2 Configuring Authentication

Authentication can be done using one of the two methods

- Using PSKs
- Using PKI

2.2.1.2.1 Authentication Using PSKs

“Pre-shared” means the parties must agree on a shared secret key that must then be predefined in the encryption devices.

The keys are configured as follows:

```
crypto isakmp policy 10
 authentication pre-share

crypto isakmp key Cisco address 100.1.1.5
crypto isakmp key Cisco address 100.1.1.9
crypto isakmp key Cisco address 100.1.1.13
```


2.2.1.2.2 Authentication Using PKI

In PKI-based deployments under the ISAKMP policy, the correct authentication method must be set and a certificate from a certificate authority (CA) must be obtained. These steps, which follow, must be repeated for all devices (GMs and KSs) in the network.

Configuring Authentication in IKE Phase 1

In IKE Phase 1, authentication must be configured to **rsa-sig**. This can be done as follows:

```
crypto isakmp policy 10
authentication rsa-sig
```

Note: **rsa-sig** is the default authentication method for an ISAKMP policy. This command does not appear in the configuration

Generating RSA Keys

Unique RSA keys must be generated on all KSs and GMs as follows:

```
KS(config)#crypto key generate rsa general-keys label pki_KS modulus 1024
The name for the keys will be: pki_KS

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

Note: RSA keys used for PKI are different than the RSA keys generated and used for KS synchronization and policy generation. The KS COOP RSA keys must be the same on all KS serving a group, while KS CA RSA keys should be unique.

Configuring the Router for the CA

All routers (KS and GM) are configured with trust points as follows:

```
crypto pki trustpoint GETVPN
   enrollment url http://24.100.100.100:80
   subject-name OU=GETVPN
      revocation-check none
      rsakeypair pki_KS
```

Authenticating to the CA Server

Authenticating to the CA server is performed to receive the certificate from the CA. The configuration follows:

```
KS(config)#crypto pki authenticate GETVPN
Certificate has the following attributes:
       Fingerprint MD5: FFD61C4E F12676BA FAADEFD4 E205EA6B
      Fingerprint SHA1: 4530D929 EA0A6383 14241669 6B7063DB D765D162

% Do you accept this certificate? [yes/no]: y
Trustpoint CA certificate accepted.
```


Enrolling to the CA Server

Requesting a router certificate from the CA is done as follows:

```
KS(config)#crypto pki enroll GETVPN
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the
   configuration.
   Please make a note of it.

Password:
Re-enter password:

% The subject name in the certificate will include: OU=GETVPN
% The subject name in the certificate will include: KS
% Include the router serial number in the subject name? [yes/no]: n
% Include an IP address in the subject name? [no]: n
Request certificate from CA? [yes/no]: y
% Certificate request sent to Certificate Authority
% The 'show crypto ca certificate GETVPN verbose' command will show the
fingerprint.

KS(config)#
May 12 15:12:43: CRYPTO_PKI:  Certificate Request Fingerprint MD5:
1F12E4B5 ABDB70F8 E0A7DB49 DD327570

May 12 15:12:43: CRYPTO_PKI:  Certificate Request Fingerprint SHA1:
18186C7B 1FFDCFAA 42678D80 1633479F 415D50EB

GroupMember-1(config)#
May 12 15:12:45: %PKI-6-CERTRET: Certificate received from Certificate
Authority


KS#sh crypto ca certificates
Certificate
  Status: Available
  Certificate Serial Number: 0x6E
  Certificate Usage: General Purpose
  Issuer:
    cn=GET
    ou=NSITE
    o=CISCO
    l=RTP
    st=NC
  Subject:
    Name: KS
    hostname=KS
    ou=GETVPN
  Validity Date:
    start date: 10:15:09 EST Apr 25 2008
    end   date: 10:15:09 EST Apr 25 2010
  Associated Trustpoints: GETVPN

CA Certificate
```



```
  Status: Available
  Certificate Serial Number: 0x1
  Certificate Usage: Signature
  Issuer:
    cn=GET
    ou=NSITE
    o=CISCO
    l=RTP
    st=NC
  Subject:
    cn=GET
    ou=NSITE
    o=CISCO
    l=RTP
    st=NC
  Validity Date:
    start date: 17:11:56 EST Jun 12 2007
    end   date: 17:11:56 EST Jun 10 2012
  Associated Trustpoints: GETVPN
```

Tip: A.3.1 "Configuration" contains a sample IOS CA configuration. It is also possible to deploy IOS CA and KS on the same device for small-scale GET VPN deployments.

2.2.2 Configuring IPsec Parameters

Transform-set and lifetime configuration for the group IPsec SA is defined as follows.

```
crypto ipsec transform-set mygdoi-trans esp-aes esp-sha-hmac

crypto ipsec profile gdoi-profile-getvpn
set security-association lifetime seconds 7200
set transform-set mygdoi-trans
```

2.2.3 Configuring GDOI Group

Before starting KS configuration, generate the RSA keys to be used during rekeys.

```
keyserver1(config)#crypto key generate rsa general-keys label
                  getvpn-export-general modulus 1024 exportable
```

Note: RSA keys must be generated on any KS. All KSs must share the same keys, so these keys must be generated with an "exportable" tag. The keys are then imported on the remaining KSs. These keys do **not** need to be imported on the GMs.

2.2.3.1 Configuring Unicast Rekeys

2.2.3.1.1 Configuration using Unicast Transport Mechanism

The following configuration enables the KS in a router. Each group defined in the KS has an identity that is shared among the GMs within the group. Here, the identity is set to 1234 for the group 'getvpn'. The KS also defines policies using access-list 199 to be distributed to GMs upon registration.

Further rekeys are sent through the unicast transport mechanism with two more retransmits at 10 second intervals. The KS uses this retransmit configuration to resend rekeys in case acknowledgements are not received from the GM for the rekey sent earlier. The lifetime validity of the rekey policy is configured for 24 hours. The time-based anti-replay (TBAR) value is set to five seconds.

```
crypto gdoi group getvpn
identity number 1234

! local keyword identified this router as key server
server local

! lifetime of rekey policy set to 24 hours
rekey lifetime seconds 86400

rekey retransmit 40 number 2

! RSA key
rekey authentication mypubkey rsa getvpn-export-general

! Rekeying through unicast transport
rekey transport unicast

sa ipsec 1

! Transform-set for group members
profile gdoi-profile-getvpn

! Policies defining traffic to be encrypted
match address ipv4 199


! Time based anti-replay set to 5 sec
replay time window-size 5

! Source address of the rekey packet
address ipv4 100.1.1.1
```

The GET VPN policies defined in the KS are downloaded to all GMs.

2.2.3.2 Configuring Multicast Rekeys

It is assumed that all multicast routing (Protocol Independent Multicast (PIM), sparse mode, rendezvous point, and so on) are configured and working before the multicast rekey configuration is deployed.

2.2.3.2.1 Configuration using Multicast Transport Mechanism

Multicast rekey is the default rekey method in GET VPN (do not configure “rekey transport unicast”).The multicast group address, to which rekeys are sent, must be configured.

```
crypto gdoi group getvpn
 identity number 1234
 server local
!
! multicast group address
!
  rekey address ipv4 198
  rekey lifetime seconds 86400
```



```
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa getvpn-export-general
  sa ipsec 1
   profile gdoi-profile-getvpn
   match address ipv4 199
   replay time window-size 5
```

The access-list is defined for any source address to access the multicast group 239.77.77.77 as follows

```
access-list 198 permit ip any host 239.77.77.77
```

2.2.4 Configuring Access List Policies

The following access control list (ACL) defines policies to be pushed to GMs. This enables only GMs that are configured in the 10.1.0.0/16 network address range to communicate using GET VPN. The ACL is mapped to the GET VPN configuration as shown in 2.2.3.1 Configuring Unicast Rekeys. It is also possible to define an ACL to encrypt all subnets in the network simply by defining a “permit ip any any” entry, and adding deny statements in the ACL to deny all control plane traffic from the encryption policies.

```
access-list 199 remark ACL policies pushed to authenticated group
                       members
access-list 199 permit ip 10.1.0.0 0.0.255.255
                          10.1.0.0 0.0.255.255
```

2.3 GM Configuration

It is assumed that all connectivity (default routes, RPs, and so on) is set up initially before GM configuration takes place

2.3.1 Configuring IKE Phase 1

IPsec transform-sets and profile configurations are not required on GMs. These parameters are pushed down by the KS as part of GDOI registration. Only ISAKMP configurations are required to enable a GM and KS to authenticate each other.

```
crypto isakmp policy 10
encr aes
lifetime 1200
authentication pre-share
group 2

crypto isakmp key Cisco address 100.1.1.1
crypto isakmp key Cisco address 100.1.1.5
```

For the PSK authentication method, PSKs are needed in each GM only to authenticate the KS. Defined PSKs are not required to authenticate other GMs. PKI configuration is the same as in the KS. See 2.2.1.2.2 Authentication Using PKI for more information about PKI configuration.

2.3.2 Configuring the GDOI Group

A GM is configured using the same group identity defined on the KS and the address of the KS.

```
crypto gdoi group getvpn
identity number 1234
```

```
! Registration with preferred key server
server address ipv4 10.1.1.1

! If preferred KS not reachable, register with alternate KS in the group
server address ipv4 10.1.1.5
```

2.3.3 Configuring Crypto Map

The crypto map has a new type, **`gdoi`**, and is tied to GDOI group created in the preceding section.

```
crypto map getvpn-map 10 gdoi
set group getvpn
```

2.3.4 Enabling GET VPN

Applying crypto map to the WAN interface enables GDOI.

```
interface Ethernet0/0
description WAN interface to MPLS PE
ip address 100.1.1.9 255.255.255.252

! crypto map enabled on WAN interface
crypto map getvpn-map
```

Note: The **`ip tcp adjust-mss`** command is used to limit maximum segment size for TCP sessions, to avoid maximum transmission unit (MTU) issues causes by IPsec overhead. It is applied to the LAN interface of the GM as follows:

```
interface Ethernet0/1
description LAN interface of GM
ip tcp adjust-mss 1360
```

2.4 COOP KS Configuration

The preceding configuration is sufficient for a standalone KS in an enterprise network. This section describes the configuration of a COOP KS. Before deploying COOP KS configurations, consider the following:

- Generate RSA keys in the primary KS (as required for rekeys) and export both private and public keys to all the COOP KSs. This is required in case the primary KS goes down; the rekeys sent by the newly elected primary KS will still be properly verified by the GM.
- Election between the KSs is based on the highest-priority value configured. If they are same, it is based on highest IP address. It is suggested to configure priorities for selecting the primary KS for easy setup and troubleshooting.
- Periodic ISAKMP keepalive (dead peer detection, or DPD) must be configured on the COOP KSs so that the primary KS can track the secondary KS state.
- Rekey configuration, policies, and anti-replay configurations must be the same for all KSs. This is not done automatically in the current phase of GET VPN. The user must manually track configurations.

2.4.1 Exporting and Importing RSA Keys

The procedure to export and import RSA keys follows:

```
Primary_Key_Server(config)#crypto key generate rsa general-keys label
                getvpn-export-general modulus 1024 exportable
% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be exportable...[OK]
```

Export this key to the terminal.

```
Primary_Key_Server(config)#crypto key export rsa getvpn-export-general pem
                terminal 3des passphrase
% Key name: getvpn-export-general
   Usage: General Purpose Key
   Key data:
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC96RhInBlxIGAq4bYd4z1FwWft
cJKAoJTxfoKYwZpi5+PZ41CApgO/8Y0SJLuXnpDVlxWbjNTIoVf4RQyerQSvph6X
BBvX4j5d9pJZJdcdIBymq3F/CEnnbJWxukHQCnN1UCgdJ87oTp4gN7THaGFM3ui2
PgfEpUH5WujPrSCQ4QIDAQAB
-----END PUBLIC KEY-----
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,5DD792A00CA3675D

nVKal5JqzMUeqF/1DQKDkTrNOeB5GSOqbm6/8/lz2Z6SoxbyRICSsk7tsntdvyo9
SxVh5nf4Z//gj4cKm9cM4DvC8ui66WncIvhU1+dCqQ9QK1OOZ1T3GcWMr5rCvkST
+XgVtZvK/3b8CyE1vMwK5GhZ8s/BkU86fO0vdnAw3X3dlCO2kTMrttHTHh5quFJi
mhgUUJJWM//Rv5k2iU3ap99FyIB+52QEKiCMYE/WmjYp8+BLcA1qLJhqTs76P/oc
NlM2XaZquMWUqmRFJyTnveuh0tL57ZQXzL8tCmSh1B59TzfFftwlcWtrihZEzP4x
Ek3dX/VECRA5X0nc9ZOmaYJUyAY6Xy3ZfHlqCd8h1wQ+AxpuOqNUYwR/54r6v25k
kl1aK7NZ3mh075qjdsEtXNWb+9I9RrvttNkTcbfeyZifv8ZP1YN5OcX1w1Z4E17P
hxo/EFs1AwNgoY0kmgnfNi6g1f9duqykLnLtD6NONrfQQAqTLe59FXKMo9oXG3TE
KpAcDKJ37CayzWeIAKekxJX4YCX7CsT2njbM2WSoHTX90vB1sm0UajATpltil2WD
3c4Lyv8oy0A6FNHJhqZJeFW3w+A0OiHeggKL5Aytoyu41I3zozvGGvhDoyjPfXmu
cxaK2FywKuzz5KYrqvUGwGIAoMR8tn4pwQs7CM+MlrVMOOM7ghLmnlEziLSEis36
b94kIPs70heSclECWJrnOCP/NRJ84p4xvtwEPDA68bzlyQOgQ5mrLLwvbIR1CGgm
l+JyTws+j9TaDzaWSbhnt2lqtTxmkRAbRGGpjzAykONulwSXk0BZ8Q==
-----END RSA PRIVATE KEY-----
```

Import this key using cut-and-paste to other KSs in the GET VPN network. The “exportable” option supports this procedure for other KSs deployed later.

```
COOP_Key_Server(config)# crypto key import rsa getvpn-export-general
                 pem exportable terminal passphrase
% Enter PEM-formatted public General Purpose key or certificate.
% End with a blank line or "quit" on a line by itself.

% Enter PEM-formatted encrypted private General Purpose key.
% End with "quit" on a line by itself.
```

In the preceding example, the terminal was used to copy RSA keys from one KS to another. A Trivial File Transfer Protocol (TFTP) server can also be used to export and import keys.

```
! To Export the key
!
crypto key export rsa <label> pem url tftp: 3des <passphrase>
!
```



```
! To Import the key
!
crypto key import rsa <label> general-purpose pem url tftp: <passphrase>
```

Tip: Exporting and importing RSA keys using a TFTP server is a more consistent and preferred method. A TFTP server can also be used to securely store keys for new KS deployments. For secure key storage, additional steps might be necessary.

2.4.2 Configuring KS Redundancy

The following configuration enables redundancy on the KSs.

2.4.2.1 Primary KS

```
! enabling ISAKMP keepalives (DPD)
!
Crypto isakmp keepalive 15 periodic
!
crypto gdoi group getvpn
server local

! enabling cooperative key server function
redundancy

! priority decides the role of this key server
local priority 100

!All other key servers must be configured
peer address ipv4 100.1.1.5
```

2.4.2.2 Secondary KS

```
!
Crypto isakmp keepalive 15 periodic
!
crypto gdoi group getvpn
server local

! enabling cooperative key server function
redundancy

! priority decides the role of this key server
local priority 75

! All other key servers must be configured
peer address ipv4 10.1.1.1
```


3 System Design

To design a GET VPN network, one starts with selection of supported platforms and Cisco IOS releases for the group members (GMs) and one or more key servers (KSs). KS selection depends largely on scalability requirements, while GM selection depends largely on performance.

After selecting the right platforms and software for a GET VPN network, choosing the right policies for KSs, cooperative (COOP) KSs, and GMs is vital for successful deployment and operation. The maximum transmission unit (MTU) in a GET VPN network is another important design consideration. This chapter addresses these design considerations in detail.

3.1 Platform Support

Table 3-1 lists the most common platforms in the Cisco enterprise portfolio having encryption capability and modules targeting GET VPN support.

Table 3-1 Currently Supported GET VPN Platforms

Product Line	GM	KS
Software	Not recommended	Not recommended
87x (onboard)	Yes	Not recommended
1800/1841 (onboard, AIM/SSL)	Yes	Yes for 1841
2800 (onboard, AIM/SSL)	Yes	Yes
3800 (onboard, AIM/SSL)	Yes	Yes
7200 NPEG1/NPEG2, VAM2+	Yes	Yes
7301 VAM2+	Yes	Yes
7201 VAM2+	Yes	Yes
7200 NPEG2, VSA	Yes*	Yes*
ASR	No	No
6500 VPN-SPA	No	No

* Time-based anti-replay (TBAR) is not supported on VSA.

3.2 IOS Software Releases

The current supported release is 12.4(15)T5.

3.3 KS Selection

KS selection depends largely on the required network scalability (the number of GMs supported in a group). The limiting factors in KS scalability are the registration rate (how quickly GMs can register with a KS) and the ability of the KS to handle rekeys to maintain GM synchronization.

By far, the registration rate is the single most important factor in the KS selection process. The goal of KS server selection for a specific network is to keep registration time low so that, in case of KS rekey failure, GMs can reregister within a reasonable time and continue to forward data without disruption.

Table 3-2 lists KS scalability numbers tested under the following conditions:

- All KSs are configured in COOP mode
- The maximum number of GMs were tested for both unicast and multicast rekeys
- All GMs registered to the KS is less than 20 seconds

For example, the first row indicates that the Cisco 7200 with a VAM2+ crypto card can support up to 2000 GMs, that the maximum CPU usage is 40% for registration and 18% for rekey, and that the time to finish registration is 15 seconds when GMs registration is distributed on two KSs.

Table 3-2 KS Scalability

Platform	Crypto Card	Max Number of GM Tested	Max Registration CPU / MAX Rekey CPU	Time to register to a single KS
7200/7201	VAM2+	2000	40/18 %	15 sec *
3845	AIM-VPN/SSL-3	1000	46/20 %	15 sec *
3825	AIM-VPN/SSL-3	500	34/14 %	15 sec
2851	AIM-VPN/SSL-2	200	25/15 %	15 sec
2821	AIM-VPN/SSL-2	100	30/14 %	15 sec
1841	AIM-VPN/SSL-1	50	16/8 %	15 sec
7200/PKI	VAM2+	1000	30/10 %	20 sec **

* GM registration was distributed over two KSs to reduce the registration time to the values shown

** GM registration was distributed over 4 KSs to reduce the registration to value shown.

Note: GM registration using PKI (the IKE setup rate) is lower than GM registration using pre-shared keys (PSKs). Since time allowed for GM registration was set to only 20 seconds, supported number of GMs using PKI was limited to 1000. If time allowed for registration was increased by a few minutes, KSs would scale the same whether using PKI or PSKs

3.4 GM Selection

A GM should be selected based on the required throughput or packet forwarding rate. If the traffic mix primarily comprises small packets (for example, as in VoIP), the packets/second performance is more important. If the traffic mix primarily comprises large packets for example, as in file transfers), throughput performance is more important. This is because the packet/second forwarding performance is processor bound, while throughput performance is typically crypto engine bound.

3.4.1 GM Performance

GM performance characterization is based on the following parameters:

- QoS is not enabled
- Hardware crypto
- Identical configuration for all platforms

There is no difference in performance whether using Advanced Encryption Standard (AES) or Triple DES (3DES). The numbers below represent the best case scenario i.e. when other services are not turned on.

For example, the first row of Table 3-3 indicates that a Cisco 871 with onboard encryption can push through the maximum of 1Mb/s clear LAN traffic for 74bytes, 7Mb/s for IMIX, and 28Mb/s for 1400bytes. The traffic rate on the WAN interface (after encryption) is a little higher because of the encryption overhead.

Table 3-3 GM Performance for Clear LAN Traffic (Mb/s)

Platform	Crypto Cards	74 Bytes	IMIX*	1400 Bytes
C871	On Board	1	7	28
C1841	On Board	2	9	33
C1841	AIM-VPN/SSL	4	22	83
C2821	Onboard	10	30	50
C2821	AIM-VPN/SSL	15	70	190
C2851	Onboard	10	46	64
C2851	AIM-VPN/SSL	16	86	190
C3825	Onboard	21	98	283
C3825	AIM-VPN/SSL	26	116	199
C3845	Onboard	27	126	284
C3845	AIM-VPN/SSL	31	140	200
C7200 NPE G1	VAM2+	35	159	266
C7200 NPE G2/7201	VAM2+	72	183	283
C7301	VAM2+	35	163	283
C7200 NPE G2	VSA	230	230	940

* IMIX pattern is 7 64-byte packets, 4 568-byte packets and 1 1400-byte packet

Note: The GET VPN throughput is similar to IPsec throughput with TBAR disabled. With TBAR enabled, throughput drops by approximately 5%. Throughput numbers in Table 3-3 were captured with TBAR enabled.

3.4.2 Number of Security Associations

Some platforms (e.g. 870, 1800) can support only a limited number of security associations (SAs). For information about the supported number of SAs, refer to:

http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6586/ps6635/ps7180/prod_brochure09186a00801f0a72_ns710_Networking_Solutions_Brochure.html

In GET VPN, like almost all other IOS based IPsec solutions, new IPSec SAs are created before existing IPsec SAs expire. Therefore, an important platform (or encryption module) selection consideration is the ability to support at least twice the number of configured IPsec SAs.

For example, if a GET VPN policy results in creation of 100 SAs on the GM (50 permit entries in the encryption ACL), the platform should be able to handle 200 or more SAs. Note that many policy changes on the KS (ACL changes, for example) also result in a rekey and therefore create additional SAs.

3.4.3 Encryption Modules

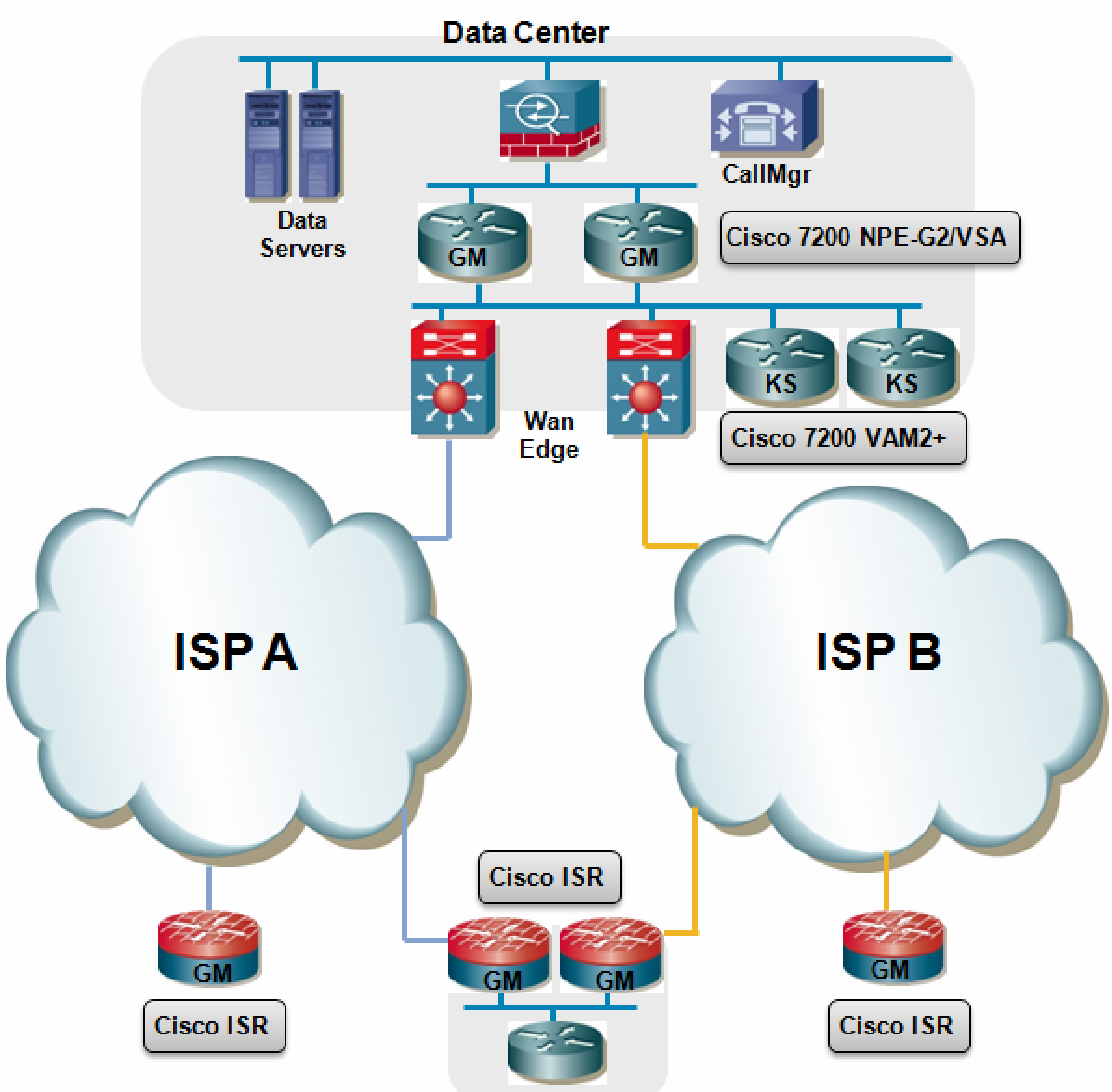
AIM/SSL cards are recommended for ISR routers, but they not required for all ISR models. ISR AIM/SSL cards perform better than on-board encryption engines in most cases. C7200 & C7300 routers can use VAM2+; VSA is used together with C7200 NPE G2.

3.4.4 Limitation of VSA

C7200 VSA does not support TBAR, an important consideration if VSA will be deployed in the network. TBAR will be unable to run for the whole network.

Figure 3-1 shows a typical choice of IOS platforms as GMs and KSs at the branch and data center (DC) based on these parameters. The choice of ISR model at branches depends on the connection type and throughput requirements, among other parameters.

Figure 3-1 Typical Enterprise Choice of Platforms



3.5 KS Design Considerations

3.5.1 ISAKMP

The KS should be configured in AES mode for encryption using 128 bit (or better) keys because AES mode provides more robust security with reduced computation overhead.

The lifetime of ISAKMP sessions on the KS is recommended to be the default lifetime of 24 hours. The KSs need the IKE session active to transmit COOP messages between themselves. This is a persistent database synchronization process, so the IKE session is always required. There is no point in tearing down the IKE session because it is immediately restored.

To configure:

```
crypto isakmp policy 10
 encryption aes
 lifetime 86400
```

To verify the preceding configuration, use “show crypto isakmp policy”.

ISAKMP periodic dead peer detection (DPD), also called keepalives, must be configured on all KSs (only) so the primary COOP server can keep track of the state of the secondary KSs.

```
crypto isakmp keepalive 15 periodic
```

3.5.2 IPsec

AES mode is recommended for the Traffic Encryption Key. Since AES mode provides more robust security with minimal computation overhead.

To configure:

```
crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
!
crypto ipsec profile gdoi1
 set security-association lifetime seconds 7200
 set transform-set AES_SHA
!
crypto gdoi group dgvpn1
 identity number 61440
 server local
  rekey lifetime seconds 86400
  rekey retransmit 40 number 3
  rekey authentication mypubkey rsa getkey
  rekey transport unicast
  sa ipsec 1
   profile gdoi1
   match address ipv4 160
   no replay
  address ipv4 139.1.1.3
```

To verify the configuration, use **`show crypto gdoi ks policy`**.

3.5.3 Traffic Encryption Key Lifetime

The traffic encryption key (TEK) lifetime should be no less than the default 3600 seconds. The TEK lifetime should **never** be set below 900 seconds in real deployments because a short TEK lifetime creates more key rollovers that must be synchronized among all GMs. If the KS has insufficient time to complete key distribution before prepositioning the security association with the next TEK, the system operates in an unstable state. The longer lifetime improves network stability and minimizes network change. A TEK lifetime of two hours (7200 seconds) is a good value to use.

If policies change frequently on the KS (for example, during early deployment stages), the TEK lifetime can be lowered to minimize the number of active SAs. This is commonly done in a lab setting where clearing the entire network is feasible.

To change the lifetime, use the following configuration:

```
crypto ipsec profile gdoi1
 set security-association lifetime seconds 7200
```

To verify the configuration, use “show crypto gdoi ks policy”.

Note: The actual time at which the KS sends rekeys depends on the number of GMs, retransmission values, and a maximum of 10% or 90 sec head start value. The following formula can calculate the actual rekey time:

Lifetime – [Max(10% of lifetime or 90 sec) + (required retransmission time) + (5 x Number of GMs/50)]

For example, for 1000 GMs, a configured TEK lifetime of 7200 seconds and two retransmissions (60 seconds apart), the KS sends rekeys every 6260 seconds:

7200 – [720 (10% lifetime) + 120 (60x2 retransmissions) + 100 (additional time to cover 1000 GMs)]

3.5.4 ACL in Traffic Encryption Policy

The permit entries in the access control list (ACL) for encryption policy should include the subnets which must be encrypted. The maximum number of lines in a traffic ACL is 100. Note that each permit statement in the KS ACL results in an SA on the GM, so the number of permit entries should be limited to minimize the SA database (SADB) on the GM.

As mentioned, it is possible to add a single “permit ip any any” entry in the ACL to encrypt all traffic. However, explicit deny entries should be configured in the ACL to exclude control traffic (for example, routing protocols) from encryption. The network designer must determine what traffic requires encryption.

The following protocols, which are commonly denied in encryption policy, are provided for reference.

```
ip access-list extended encryption_list
 deny   esp any any
 deny   tcp any any eq tacacs
 deny   tcp any eq tacacs any
 deny   tcp any any eq ssh
 deny   tcp any eq ssh any
 deny   tcp any any eq bgp      ; when GM's use BGP for PE-CE adjacency
 deny   tcp any eq bgp any      ; when GM's use BGP for PE-CE adjacency
 deny   ospf any any            ; when GM's use OSPF for PE-CE adjacency
 deny   eigrp any any          ; when GM's use EIGRP for PE-CE adjacency
 deny   pim any 224.0.0.0 0.0.0.255
 !deny   udp any any eq ntp     ; optional
```



```
 !deny   udp any any eq dns     ; optional
 !deny   udp any any eq snmp    ; optional
 !deny   udp any any eq syslog  ; optional
 !deny   udp any any eq 1645    ; optional
 !deny   udp any any eq 1646    ; optional
 !deny   udp any any eq 1812    ; optional
 !deny   udp any any eq 1813    ; optional
 !deny   tcp any eq 443 any     ; optional
 !deny   tcp any any eq 443     ; optional
 deny   udp any eq isakmp any eq isakmp
 deny   udp any any eq 848
 permit ip any any
```

Note: One should exercise extreme caution when using **`permit ip any any`** entry during deployment. If this entry accidentally precedes deny entries for control and management traffic, and the ACL is downloaded to GMs, this can break control/management traffic by encrypting updates. One must do a **`clear crypto gdoi`** on each GM, or wait until the SA times out.

3.5.5 Key Encryption Key Lifetime

The key encryption key (KEK) lifetime should be left at the default of 86400 seconds. Because the KEK is used to encrypt control plane messages between the KS and GMs, frequently changing the KEK value subjects GMs to possible rekey misses and subsequently requires GMs to reregister more frequently than necessary.

To change the value:

```
crypto gdoi group dgvpn1
 identity number 61440
 server local
  rekey lifetime seconds 86400
```

To verify the value, use **`show crypto gdoi ks policy`**.

3.5.6 Rekey Retransmit Interval

Rekey retransmits should be configured using one of the following schemes:

- Two retransmissions at 60 second intervals or
- Three retransmissions at 40 second intervals.

The rekey retransmit interval should be set as shown because for groups with many GMs, it can take a significant time to send unicast rekey messages. Retransmission should start after a complete cycle of rekey messages with sufficient time for acknowledgements.

Larger rekey intervals and retransmissions ensure that TEKs are prepositioned well in advance of TEK expiration, and mitigate the impact of network partitioning. The retransmission and rekey interval duration should also exceed the KS DPD, and IP convergence in the core network.

To configure this value:

```
crypto gdoi group dgvpn1
```



```
 identity number 61440
 server local
  rekey retransmit 40 number 3
```

To verify this value, use s**`how crypto gdoi ks rekey`**.

3.5.7 TBAR

On all platforms except C7200 NPE G2/VSA 12.4(15)T5, TBAR should be configured and counter-based anti-replay should **not** be configured. VSA does not support TBAR in 12.4(15)T5, but TBAR will be supported in future releases.

Because multiple sources and destinations are using the same SA, counter-based anti-replay makes no sense and TBAR is the only viable method. TBAR is sufficient because the KS maintains time synchronization for all GMs. This is not global time; but is a relative time clock for that security group. No Network Time Protocol (NTP) or GPS based time synchronization is required. The KS periodically transmits time sync rekeys before traffic encryption key (TEK) expiration.

By default, counter-based anti-replay is configured. To configure TBAR:

```
crypto gdoi group dgvpn1
 identity number 61440
 server local
<..>
  sa ipsec 1
<..>
  replay time window-size 5
```

To verify this configuration, use “show crypto gdoi”.

3.5.8 Authentication Policy for GM Registration

GMs can authenticate to the KS at registration time using PSKs or PKI. PSKs are easy to deploy but must be managed proactively. It is recommended to deploy a peer-based PSK instead of defining a default key (the key defined with an address of 0.0.0.0) for all the devices in the network. PSKs should be updated regularly (every few months).

Note: A PSK can be updated on a KS-GM peer basis without affecting the crypto data plane or control plane because rekeys are secured using the KEK. It is important to ensure that a GM can reregister to each ordered set of KSs using the newly assigned PSK.

PKI uses its infrastructure to overcome the key management difficulties encountered when using PSKs. The PKI infrastructure acts as a certificate authority (CA) where router certificates are issued and maintained. However, using PKI during IKE authentication is computationally intensive. In PKI deployments, KS capacity, design, and placement become very important.

For added security, GET VPN also supports GM authorization as described in the following sections.

3.5.8.1 GM Authorization Using PSKs

GET VPN supports GM authorization using the **IP address** when using PSKs. An ACL matching the WAN addresses (or subnets) of the GM can be defined and applied to the GET VPN group configuration. Any GM whose IP addresses match the ACL authorizes successfully and can register to the KS. If a GM IP address does not match the ACL, KS rejects the GM registration request.

```
!
! On KS
!
crypto gdoi group getvpn
 server local
  authorization address ipv4 50
!
access-list 50 permit 100.1.1.10
access-list 50 permit 101.1.1.0 0.0.0.255
!
```

In cases of unsuccessful authorization, the following syslog message is generated:

```
%GDOI-1-UNAUTHORIZED_IPADDR: Group getvpn received registration from
unauthorized ip address: 100.1.1.9
```

3.5.8.2 GM Authorization using PKI

GET VPN supports GM authorization using the **DN** (common) or **FQDN** when using PKI. A crypto identity matching certain fields in the GM certificate (typically OU) can be defined and applied to the GET VPN group configuration.

Any GM whose certificate credentials match the ISAKMP identity is authorized and can register to the KS. For example, if all GM certificates are issued with OU=GETVPN, a KS can be configured to check (authorize) that all GMs present a certificate having OU=GETVPN. If the OU in the certificate presented by a GM is set to something else, the GM will not be authorized to register to the KS.

```
! On KS
!
crypto isakmp identity dn
!
crypto pki trustpoint GETVPN
<SNIP>
 subject-name OU=GETVPN
!
crypto gdoi group getvpn
 server local
  authorization identity GETVPN_FILTER
!
crypto identity GETVPN_FILTER
 dn ou=GETVPN

! On GM
!
crypto isakmp identity dn
!
crypto pki trustpoint GETVPN
<SNIP>
 subject-name OU=GETVPN
!
```


If authorization is unsuccessful, the following syslog message is printed on the KS:

```
%GDOI-1-UNAUTHORIZED_IDENTITY: Group getvpn received registration from
unauthorized identity:  Dist. name: hostname=GroupMember-1,ou=TEST
```

Tip: It is a best practice to turn on GET VPN authorization. When a KS serves multiple GDOI groups, KS authorization is required to prevent a GM in one group from requesting keys and policies from another group. The ISAKMP authorization confirms the GM is allowed to request GDOI attributes from the KS while the GDOI authorization confirms the GM is allowed to request GDOI attributes from a specific group configured in the KS.

3.5.9 Rekey Transport

Unicast and multicast are the two methods of transport for GDOI rekey services. Unicast rekey is required when the WAN infrastructure does not support multicast or when the operator wishes to have positive acknowledgement of GM participation in the group. Multicast rekey is required to scale GET VPN networks beyond the limits of a KS platform's unicast rekey capabilities. For supporting multicast rekeys, the WAN infrastructure must support multicast otherwise, GM's will persistently reregister.

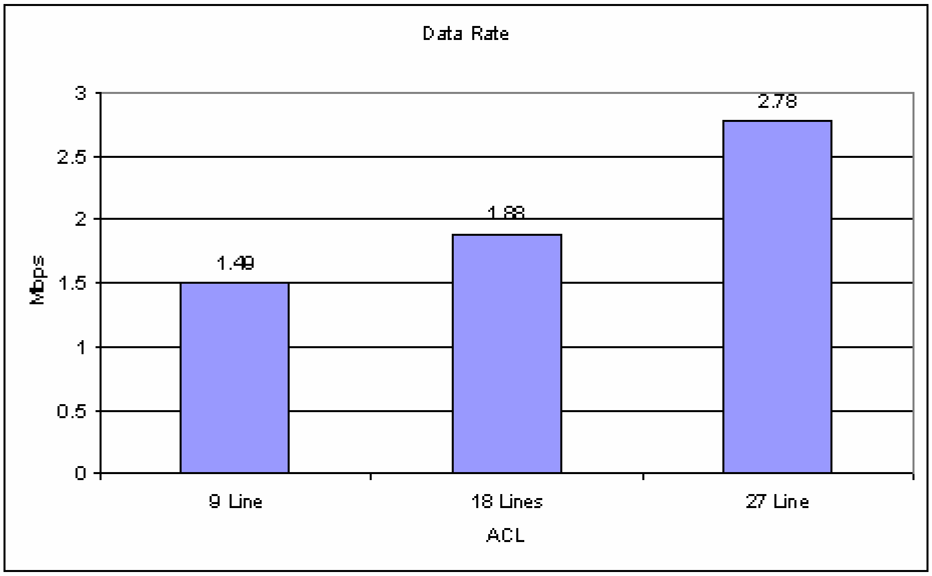
Tip: The rekey messages should be sent from an IP address that is not affiliated with a physical interface on the KS. Should the KS lose an interface, the routing control plane can reconverge and provide rekey messages via an alternate interface while using the same source IP address. A loopback interface is recommended as the source of the rekey messages.

3.5.9.1 Unicast Rekey

GET VPN rekey via unicast transport is recommended for deployment since it is more reliable method. The GM responds with a rekey acknowledgement and the KS processes the acknowledgment for each rekey it sends out. The KS will attempt to resend if no acknowledgment is received from a particular GM. The KS will remove the GM from the KS database after 3 failed rekey retransmits.

The KS can also be a bottleneck in scaling to a very large GET VPN (1000 or more GMs). While registration rate is one important factor, another important factor is the traffic burst during rekeys. It is worth noting that even for 1000+ GMs, this burst lasts only for a few seconds (10-20 secs). Below is a comparison chart for the traffic burst during rekey for different ACL sizes. From this chart, we can see that traffic rate increases with increased ACL size. Network designer should consider allocating enough bandwidth for KS in a GET VPN network.

Figure 3-2 Unicast Rekey Data Rate and ACL Size



3.5.9.2 Multicast Rekey

GET VPN rekey transport using multicast is more efficient and scalable, however, currently there is no acknowledgement for each rekey a KS sends out. The KS transmits a multicast rekey message that is delivered via the multicast infrastructure built in the WAN. Because the rekey is a single message, rekey loading is not a factor in scaling bandwidth or KS platform performance.

The KS must still be scaled to handle the 'flash' registration load should the multicast rekey fail to reach all GMs. The registration load can be distributed among several KS acting as COOP servers in a specific group. Deploying multicast rekey requires a very robust underlying multicast network design to make multicast rekey reliable.

The multicast rekey architecture should have a multicast control plane infrastructure established that is distinct from the control plane that serves data plane multicast. Three viable multicast architectures serve the GDOI control plane:

1. PIM Source Specific Multicast (PIM-SSM)
2. PIM Sparse Mode with Rendezvous Point Auto Discovery (PIM-SM)
3. PIM Sparse Mode with Anycast Rendezvous Point and Multicast Source Discovery Protocol (PIM-SM with MSDP)

Note: Deploying multicast rekey without the fix for CSCso45508 (Multicast rekey not properly reassembled on GMs) is not recommended.

3.6 GM Design Considerations

3.6.1 ISAKMP

The AES mode of encryption with 128 bit (or better) keys are recommended for ISKAMP policy, since it provides much more robust security with minimal computational complexity compared to 3DES while DES provides minimal security capabilities.

To configure:

```
crypto isakmp policy 10
 encr aes
```

3.6.2 ISAKMP Policy Lifetime

While all the ISAKMP policy parameters must match on GM and the KS, the IKE lifetime values should be set different on both devices. IKE lifetime should not be more than 30 minutes and no less than 15 minutes. The recommended lifetime is 20 minutes or 1200 seconds. Once the GM registers with the KS, the GM will use the downloaded KEK to decrypt rekey messages and the IKE session is no longer needed to maintain the GM membership in the group. Theoretically, the GM will never need to reregister as long as rekey messages are received. Retaining the IKE session is unnecessary.

To configure:

```
crypto isakmp policy 10
 lifetime 1200
```

3.6.3 Local Deny Policy

In addition to the traffic policy configured at the KS, local policies can also be configured at the GM and added to the crypto map. This can be helpful if additional control or management traffic must be excluded from the encryption policy at that specific GM. Defining local policies on the GM, reduces the size of the policy pushed from the KS.

Consider the scenario in which most GMs use EIGRP as the routing protocol; a ‘deny ip eigrp any any’ policy could be defined at the KS to be pushed down to all GMs. However, if a few GMs use OSPF, the routing control plane at these GMs would fail to maintain an adjacency because OSPF packets would be encrypted by the GM and dropped by the adjacent routing device such as a PE (typically, PE does not participate in encryption).To prevent OSPF packets from getting encrypted, a local ACL policy can be created at the GM, as shown below. This policy will only affect the GM on which it is defined.

The crypto policy applied at the GM represents a concatenation of the KS policy with the GM policy. The order of operations is such that the locally defined GM policy is checked first, followed by the KS downloaded policy:

1. GM local deny policies
2. KS downloaded deny policies
3. KS downloaded permit policies


```
!
ip access-list extended no-encrypt-ACL
 deny   ospf any any
!
crypto map dgvpn 10 gdoi
 set group dgvpn
 match address no-encrypt-ACL
!

GM1#sh cry gdoi gm acl
Group Name: dgvpn
 ACL Downloaded From KS 139.1.1.3:
   access-list  deny eigrp any any
   access-list  deny udp any any port = 500
   access-list  deny udp any any port = 848
   access-list  permit ip any any
ACL Configured Locally:
  Map Name: dgvpn
   access-list no-encrypt-ACL deny ospf any any
```

Note: Only deny statements can be added locally at the GM. Permit statements are not supported in the locally configured policies. In case of a conflict, local policy overrides the policy downloaded from the KS

3.6.4 ACL for Fail-Closed Encryption Mode

Fail-close means that no clear traffic will leak out of a router interface after a router reboot or an event such as "clear crypto gdoi";in this case the GM has not contacted the KS to download policies. Fail-open means the GM can forward any traffic in cleartext until the downloaded crypto policy is applied following registration.

A GM boots in Fail-open mode and remains in this mode until the GM successfully registers to a KS. To change this default behavior and make a GM behave in a Fail-close manner, an ACL should be used to block traffic that should be encrypted while the GM has no keys or policy facilitating encryption. The ACL typically is the inverse of the ACL defining the encryption policy. The ACL must also explicitly allow encrypted traffic through the router. This outbound ACL, however, does not block traffic originating from router itself.

For example, if an ACL from KS for encryption is:

```
access-list 160 deny   eigrp any any
access-list 160 deny   udp any any eq isakmp
access-list 160 deny   udp any any eq 848
access-list 160 permit ip any any
```

A fail-close ACL should be defined and applied as follows:

```
access-list 161 permit   eigrp any any
access-list 161 permit   udp any any eq isakmp
access-list 161 permit   udp any any eq 848
access-list 161 permit   esp any any
access-list 161 deny ip any any

interface Serial0/0/0
 description WAN interface
 ip access-group 161 out!
```


Tip: An ACL does not apply to locally generated traffic, which means we do not need explicit “permit” statements for control plane traffic originated at the router.

3.7 COOP Design Considerations

COOP KSs provide redundancy to GET VPN. Multiple KSs are supported by GET VPN to ensure redundancy, high availability (HA), and fast recovery in case of network failure.

The primary KS is responsible for creating and distributing group policy. It also periodically sends out group information updates to all other KSs to keep those servers in synchronization. If the secondary KSs somehow miss the updates, they contact the primary KS to directly request information updates. The secondary KSs mark the primary KS as unreachable (that is, "dead") if the updates are not received for an extended period of time.

Cooperative GDOI KSs can jointly manage the GDOI registrations for the group, which achieves load balancing during GM registration process. When a new policy is created on a primary KS, the primary KS to distributes rekey messages to GDOI GMs regardless of which KS a GM is registered with.

Note: IOS supports COOP configuration on per-group basis which means theoretically a KS can peer with different COOP KSs for different groups. A KS can be a primary for one group but secondary for a different group.

3.7.1 COOP Messages

COOP KSs use announcement messages to communicate with each other. These messages are exchanged on UDP port 848, as defined for GDOI. All KS-to-KS messages are secured using Phase I (ISAKMP) negotiated keys.

Primary KSs periodically send announcement messages to the secondary KSs. These messages enable the KSs to exchange state information about GMs and policies. The various components of these messages are:

- KS sender priority

  This value describes the priority of the sender, which is configurable using the CLI. The KS with the highest priority becomes the primary KS. If two KSs have the same priority, the KS with the highest IP address becomes the primary KS.

- KS role

  This value describes the role of a KS (primary or secondary).

- Group policies

  Group policies are maintained for a group and include information such as GM information and IPsec SAs and keys.

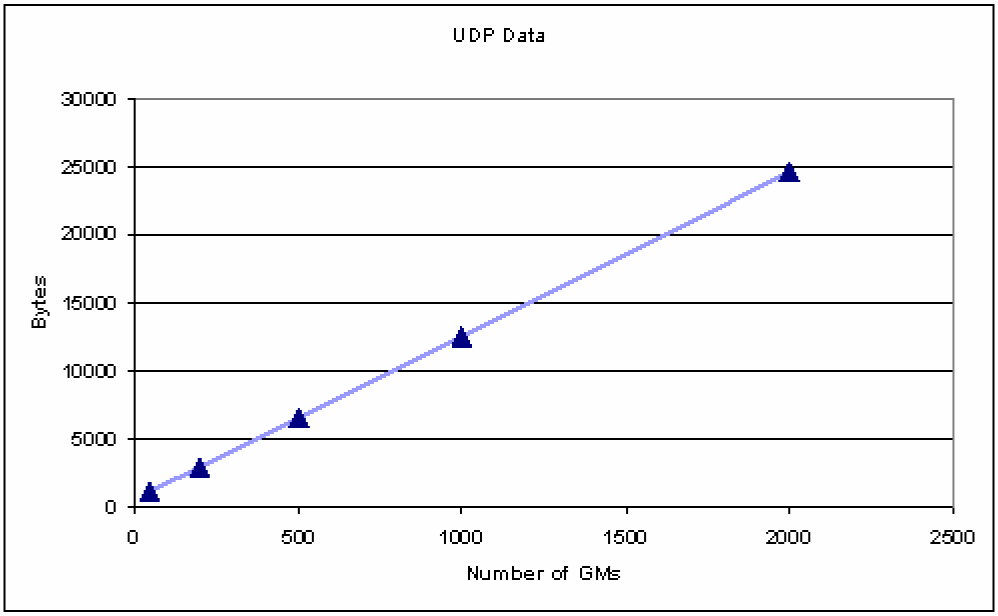
3.7.1.1 COOP Message Size

The announcement messages list GMs and communicate group policies. The number of GMs in the system can significantly impact message size. Message size also depends on the number of SAs distributed in group policies

If the number of GMs in the system is quite large, the KS system buffers should be large enough to accommodate the large messages. This can be done by increasing huge buffer size on the KSs. See 3.7.3.7 for recommendations for increasing the KS buffer size to handle large messages.

To understand the impact on the UDP payload size with increasing number of GMs, see Figure 3-3.

Figure 3-3: COOP Message Size by Number of GMs

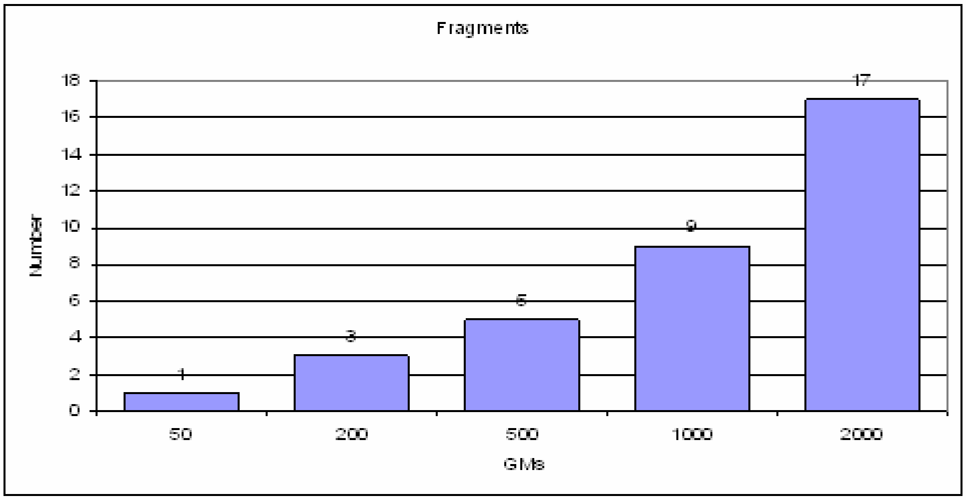


For a fixed size group policy configured on the KS (nine lines of ACL – eight denies and one permit), Figure 3-3 shows how message size increases with the number of GMs. For 2000 GMs, the size of the update message (UDP payload only) is approximately 25000 bytes.

Large message sizes mean that messages undergo fragmentation. The encryption module at the KS must be able to handle the large number of fragments. Using the same test configuration, Figure 3-4 shows how fragments increase with the increase in number of GMs. For 2000 GMs, for example, the message is contained in 17 fragments.

Note: If a 7200 NPE-G2/VSA is deployed as a GM in front of a KS, the maximum number of fragments that VSA/GM will allow is 16. Refer to CSCsm83221for more details.

Figure 3-4: COOP Message Fragments by Number of GMs



3.7.2 COOP States and Timers

COOP KSs maintain various state and role information about peer KSs.

3.7.2.1 Local KS Information

A COOP KS can operate in either a **primary** or **secondary** role. In a given GDOI group, only one KS can operate in a primary role. This assumes that all COOP KSs can exchange announcement messages and perform a role election. The role election is based on the priority value configured on each KS. 3.7.3.2 “Setting KS Priority” describes priority configuration and some design considerations for selecting the local KS priority.

To see the current role of a KS, use the following **show** command. The following examples show typical differences in priority and role.

For a primary KS:

```
KS1#show crypto gdoi ks

<..>
        Local Priority         : 100
        Local KS Status        : Alive
        Local KS Role          : Primary
```

For an operational secondary KS:

```
KS2#show crypto gdoi ks

<..>
        Local Priority         : 50
        Local KS Status        : Alive
        Local KS Role          : Secondary
```


3.7.2.2 Peer KS Information

Along with information about its own state, a COOP KS maintains role and status information for peer KSs. This information is exchanged through announcement messages (see 3.7.1 “COOP Messages”). In normal operation, COOP messages are sent only from the primary KS to the secondary KSs. The primary KS has no COOP mechanism to obtain the status of secondary KSs. Periodic DPDs must be configured on all KSs in the GET VPN network so the primary KS can correctly identify secondary KS states. Periodic DPDs are not supported between GMs and KS.

3.7.2.2.1 Peer Status

A primary KS accurately shows the status of peer secondary KSs after KS role negotiation. For a primary KS, the peer status field can have the following values:

- Alive: A peer KS, operating in secondary role, is operational
- Dead: A peer KS is unreachable

A secondary KS does **not** report the state of another secondary KS. It tracks only the status of the primary KS (either Alive or Dead). All other non-primary peer KS are reported as having ‘unknown’ status.

Note that when a KS initializes, it comes up as a secondary KS. Hence, even a KS that will become the primary KS, initially reports other peer KSs as ‘unknown’.

Tip: Secondary KS shows all non-primary peer KSs status as *unknown*.

The following example shows peer KS states as reported by a primary KS for two secondary KSs.

```
KS1#show crypto gdoi ks coop

<..>
        Peer Sessions:
        Session 1:
                Server handle: 2147483651
                Peer Address: 139.1.1.3
                Peer Priority: 50
                Peer KS Role: Secondary , Peer KS Status: Alive
                Antireplay Sequence Number: 15

                IKE status: Established

<..>
        Session 2:
                Server handle: 2147483652
                Peer Address: 139.1.1.4
                Peer Priority: 85
                Peer KS Role: Secondary , Peer KS Status: Alive
                Antireplay Sequence Number: 3345

                IKE status: Established

<..>
```

3.7.2.2.2 Peer Role

A KS reports the peer KS role as either primary or secondary. There can be only one active (‘Alive’) primary KS in a given GDOI group, assuming all KS can communicate with each other. As seen in the preceding example, a primary KS reports the peer KS role of two secondary KSs.

If a KS reports more than one primary KS, a failure condition is indicated. Consider a scenario with three COOP KSs, where KS1 is the primary KS, and KS2 and KS3 are the secondary KSs. If KS2 were to lose communication with the active primary KS (KS1) and the second secondary KS (KS3) is elected as the primary KS, the secondary KS (KS2) would report two primary KSs. However, one would be ‘Alive’ and the other would be ‘Dead’. See the following IOS snippet:

```
KS2#sh cry gdoi ks coop

<..>
        Peer Sessions:
        Session 1:
                Server handle: 2147483651
                Peer Address: 139.1.1.1
                Peer Priority: 1
                Peer KS Role: Primary   , Peer KS Status: Dead
                Antireplay Sequence Number: 27447

                IKE status: Failed
<..>

        Session 2:
                Server handle: 2147483652
                Peer Address: 139.1.1.4
                Peer Priority: 85
                Peer KS Role: Primary   , Peer KS Status: Alive
                Antireplay Sequence Number: 30

                IKE status: Established
<..>
```

3.7.2.3 Peer KS Role Transitions

A secondary KS can transition to a primary KS state under the following circumstances:

- During a COOP election process where no KS has declared itself primary KS for the group, if a secondary KS detects no other higher priority KS, that KS transitions to primary KS role
- If a secondary KS stops receiving announcement messages from the primary KS, the secondary KS transmits an announcement message to all known peers. If no responses are received, the KS transitions to the primary KS role. This is commonly seen when the network is partitioned so that one or more KSs are isolated (see 3.7.4.2 “Network Split and Merge”).

After a peer KS is elected as a primary KS, very few scenarios could cause it to switch to a secondary role. Except for initialization, the only scenario that could cause such a downgrade is detection of a higher priority primary KS. This is commonly seen after a network partition is resolved (see 3.7.4.2 “Network Split and Merge”).

A COOP KS can assume the secondary role under the following circumstances:

- Upon initialization (for example, after a reload, a KS always comes up in secondary state).
- During a COOP election process, a secondary KS that detects a higher priority KS retains its secondary state.
- If a higher priority KS in secondary role detects a lower priority KS in primary role, the higher priority KS will not preempt, and retains its secondary state.

Note: To force the KS failover, Embedded Event Manager (EEM) can be used.

3.7.2.4 COOP Timers and Parameters

This section describes COOP timers and parameters. The default values of the timers are also the recommended values. **In most cases, these timer values should not be changed**.

In addition to the COOP timers described in this section, the **ISAKMP SA lifetime** timer is also important in the COOP scenario. See 3.5.1 “ISAKMP” for more details and configuration information.

3.7.2.4.1 Primary Refresh Timer

Default: 20 seconds

The primary KS sends an announcement message to all active secondary KSs every interval defined by this timer.

To configure this timer:

```
!
crypto gdoi group <GroupName>
  server local
    redundancy
      protocol timeout refresh <seconds>
!
```

3.7.2.4.2 Secondary Periodic Timer

Default: 30 seconds

If a secondary KS does not receive a periodic announcement message from the primary KS during this interval, the secondary KS sends announcement messages with a return flag, to all KSs in the group.

To configure this timer:

```
!
crypto gdoi group <GroupName>
  server local
    redundancy
      protocol timeout periodic <seconds>
!
```

3.7.2.4.3 Retry Count

Default: 2

This parameter specifies the number of times that a secondary KS sends an announcement message to all KSs when it does not receive a periodic message from the primary KS. After this count, the secondary KS reevaluates its role based on the replies it receives.

To configure this parameter:

```
!
crypto gdoi group <GroupName>
  server local
    redundancy
      protocol retransmit <count>
```


3.7.2.4.4 Policy Update Timer

Fixed period: 5 seconds (non-configurable)

When a primary or secondary KS has a change in GM registration information, the KS must send a policy update message. This timer starts after a change in GM registration is detected. If other changes occur during this interval, all updates are sent together in one announcement message.

3.7.3 COOP KS Design Considerations

There are several design considerations related to COOP: the number of KSs, KS priority, load balancing, and more. This section describes these design considerations in more detail.

3.7.3.1 Selecting the Number of COOP KSs

Even in a multiple COOP KS setup, only the primary KS is responsible for sending out rekeys. A single KS, although sufficient, can be a single point of failure in the system. It is recommended to have at least 2 COOP KSs. It is also recommended to use as few KSs as necessary to scale the network and provide control plane resiliency.

As the number of GMs increase, it might become desirable to place KSs in geographically dispersed locations, such as different data center (DC) sites. One way to arrange or place the KSs is to have 2 KSs on a single site, and to place 1 other KS at a different geographical site. This type of placement provides redundancy for different failure scenarios.

It is recommended to place the KS in parallel with the GM as opposed to behind the GM because a GM failure can block the access to the KS. However, placing a KS behind a GM is feasible if policy exceptions are addressed to ensure reliable COOP and GDOI protocol exchanges.

The maximum number of COOP KSs in a group is seven. Note that having so many KSs in one group does not provide much benefit other than redundancy and registration load balancing. For this guide, four COOP KSs were tested.

To configure multiple COOP KSs, use the following configuration on a given KS:

```
!
crypto gdoi group <GroupName>
  server local
    redundancy
      peer address ipv4 <peerKS2>
      peer address ipv4 <peerKS3>
      peer address ipv4 <peerKS4>
!
```

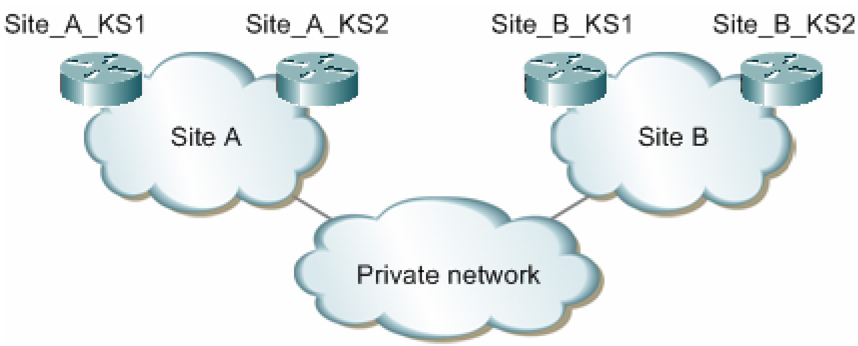
3.7.3.2 Setting KS Priority

COOP role election is based on the local priority configured on each KS. The KS having the highest priority is elected as the primary KS and all other KSs operate in secondary mode. If the KS with the highest priority fails, the secondary KS having the next highest priority takes over. The order in which a KS can assume the primary role is important.

By default, the local priority on a KS is 1. If multiple KSs have the same priority, the KS with the highest IP address wins the election.

To explain how priorities can be chosen, consider a scenario with two sites, Site A and Site B, as shown in Figure 3-5. Assume that each site has two KSs, so four KSs operate cooperatively in the same group.

Figure 3-5: KS Priority Selection Scenario



If resiliency against failed hardware or local site network failure (such as LAN segments) is most critical, KS priorities can be chosen as follows:

site_A_KS1 > site_A_KS2 > site_B_KS1 > site_B_KS2

In this case, with Site A as the primary site, site_A_KS1 is the first choice for primary KS. However, it is more likely for the local LAN segment of this KS or KS hardware to fail. Therefore, it is preferable to have site_A_KS2 as the next choice for primary.

In some systems, resiliency against site failures is most critical. In this case, priorities for the KS can be chosen as follows:

site_A_KS1 > site_B-KS1 > site_A_KS2 > site_B_KS2

In this case, Site A can be the primary site, with site_A_KS1 as the first choice for primary KS. If site A fails, it would be desirable to have a KS from site B as the next choice for primary.

The order of priorities depends on user requirements and can be chosen in many different ways. These values should be selected after careful thought and consideration.

To configure the KS local priority, configure as shown for each COOP KS.

```
!
crypto gdoi group <GroupName>
  server local
    redundancy
      local priority <1-255>
!
```

3.7.3.3 Balancing GM Registrations among COOP KSs

The primary KS is responsible for distributing rekeys to the entire system. However, GMs can register with any KS in given GDOI group. Because COOP KSs periodically synchronize their policy database, this gives load-balancing ability to the KSs.

Whenever a KS (primary or secondary) receives a new GM registration, the KS sends an announcement message with policy information for the new GM to the other KSs. This way, all KSs maintain the same complete GM database.

Load balancing of GMs can be achieved in the following ways:

- Load-balancing using configuration
- Load-balancing using routing
- Load-balancing using server load balancing (SLB)

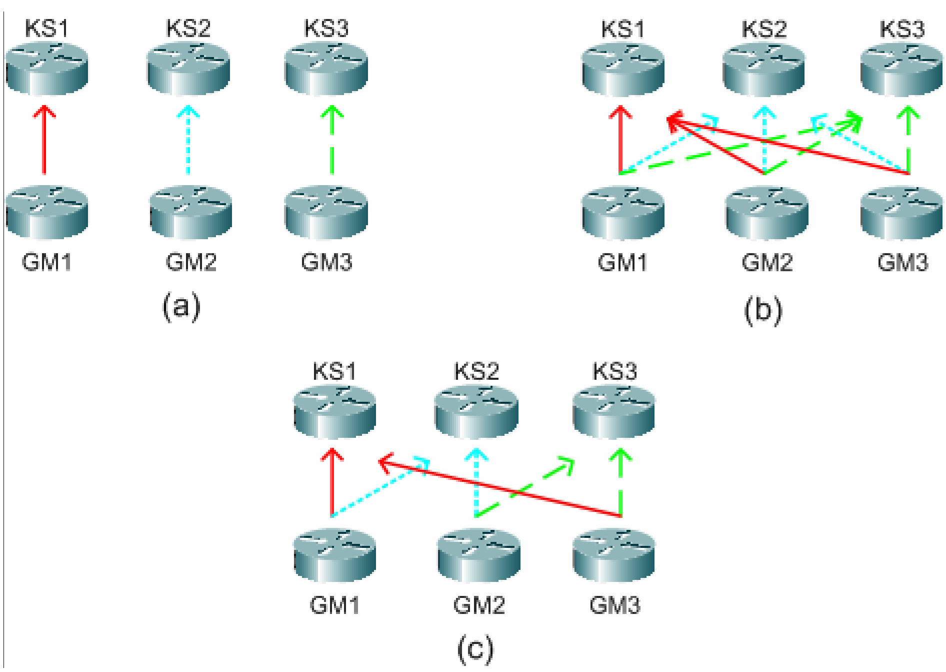
3.7.3.3.1 Load Balancing using Configuration

One set of GMs can be configured to register with KS1, another set with KS2 and so on. This provides load-balancing for registration messages. However, if one KS fails, the associated set of GMs can lose connectivity, as shown in Figure 3-6 (a).

To prevent this, each GM can be configured with all KSs. Each GM then attempts to register with the KSs in order of the configuration. If a GM cannot reach the first configured KS, it tries the next in the list, and so on. Load-balancing can be configured so that one set of GMs has KSs configured in the order KS1, KS2, KS3, KS4. Another set of GMs has KSs configured in the order KS2, KS3, KS4, KS1. In this way, all KSs are available to all GMs, but if all KSs are operational, we can have load-balancing and redundancy, as shown in Figure 3-6 (b).

Load balancing can also be accomplished using a combination of these scenarios: one group of GMs is configured with KS1 and KS2, while another set of GMs is configured with KS2 and KS3 and so on, as shown in Figure 3-6 (c).

Figure 3-6: Load-Balancing GMs on Multiple COOP KSs.



To configure multiple KSs on a GM:

```
!
crypto gdoi group <GroupName>
  server address ipv4 <KS1_IpAddress>
  server address ipv4 <KS2_IpAddress>
  server address ipv4 <KS3_IpAddress>
  server address ipv4 <KS4_IpAddress>
!
```

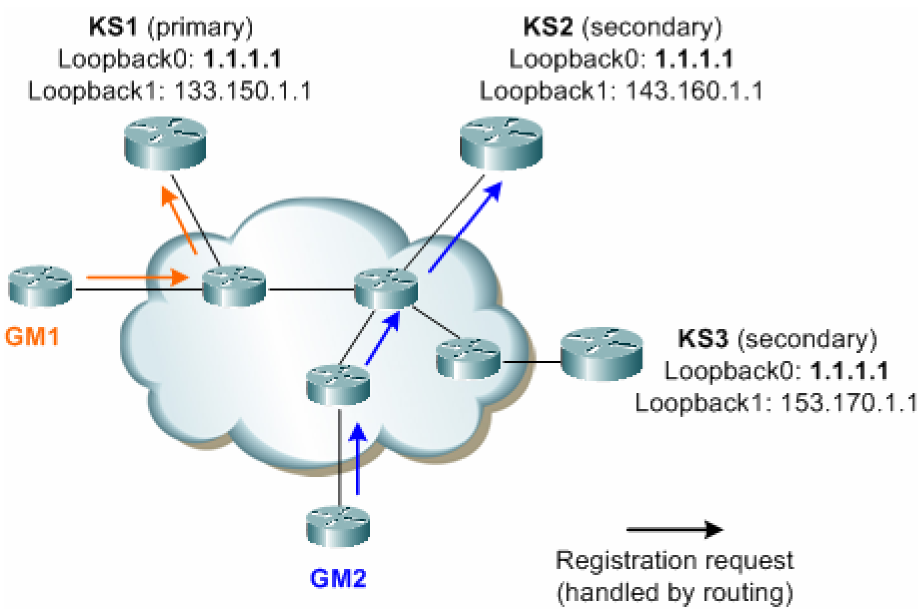
3.7.3.3.2 Load Balancing using Routing (Anycast)

If each KS shares the same registration IP address, a GM trying to reach that IP address would automatically be routed to the nearest KS. In this scenario, the load-balancing function is passed to the routing. The advantage of this method is that GM configuration is simplified; only one KS IP address must be configured at the GM. At the same time, load-balancing can only be controlled via routing. Different combinations of load balancing shown in Figure 3-6 might not be possible using this method.

Routing should be designed carefully so that each GM can reach all of the KS loopback interfaces.

Figure 3-7 shows how load-balancing is achieved using routing. As shown, all KSs have the same IP address on a loopback interface. As this loopback address is distributed to the network, routing determines the closest KS based on routing policies.

Figure 3-7: Load Balancing GM Registrations using Routing



On the GMs, only one KS must be configured.

```
!
crypto gdoi group GETVPN
 server address ipv4 1.1.1.1
!
```

On the KSs, two loopback interfaces must be configured. One loopback interface has the same IP address across all KSs used specifically for registration, while the other loopback interface has distinct IP addresses for the COOP KS functionality and for responding to registration messages. The following configuration is from KS1 as depicted in Figure 3-7.

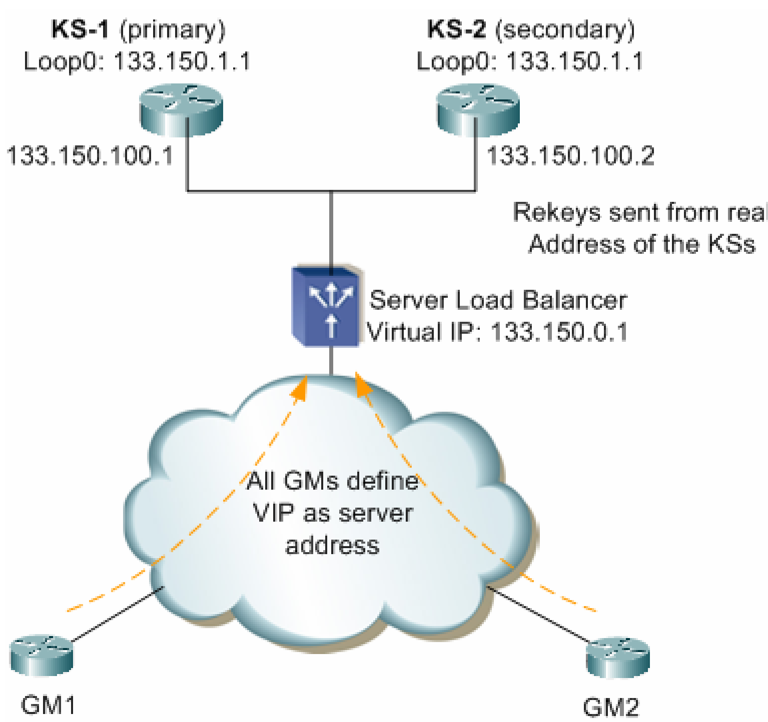
```
!
! Common IP address
interface Loopback0
 ip address 1.1.1.1 255.255.255.255
!
! Distinct IP address
interface Loopback1
 ip address 133.150.1.1 255.255.255.255
!
crypto gdoi group GETVPN
 server local
 address ipv4 133.150.1.1
 redundancy
   peer address ipv4 143.160.1.1
   peer address ipv4 153.170.1.1
!
```

3.7.3.3.3 Load Balancing using SLB

SLB provides yet another method for load balancing. If redundant COOP KSs are placed at a site, an SLB device can load-balance registrations to multiple KSs accessed using a single virtual IP address (VIP). The GMs are aware only of this VIP address, which simplifies configuration and transfers the load balancing functionality to a separate device.

Figure 3-8 shows how an SLB device can achieve load balancing. As shown, all KSs have the same IP address on a loopback interface, which is also configured on the SLB device as a virtual IP address (VIP).

Figure 3-8: Load Balancing GM Registrations Using SLB



On the GMs, only one KS must be configured.

```
!
crypto gdoi group GETVPN
 server address ipv4 133.150.1.1
!
```

On the SLB device, such as a Catalyst 6500 employing Cisco IOS SLB, a server farm is created that identifies the unique IP addresses of the KSs as “reals”. A virtual server is created for GDOI protocol and is assigned the same IP address as the common loopback IP address on the KSs.

```
!
!define a probe mechanism for the KSs
ip slb probe PING-PROBE ping
 faildetect 3
!
!define a server farm to identify the KSs
ip slb serverfarm 7200-FARM
 predictor leastconns
 failaction purge
 probe PING-PROBE
 !
 !KS-1
 real 133.150.100.1
  weight 1
  maxconns 500
  inservice
 !
 !KS-2
 real 133.150.100.2
  weight 1
  maxconns 500
  inservice
!
!define a virtual server for GDOI protocol (UDP 848)
ip slb vserver GDOI
 virtual 133.150.1.1 udp 848
 serverfarm 7200-FARM
 no advertise
 idle 900
 inservice
!
```

On the KSs, use the unique IP addresses to define the local server and peer KSs. The following configuration snippet is for KS1 as shown in Figure 3-8.

```
!
crypto gdoi group GETVPN
server local
  address ipv4 133.150.100.1
  redundancy
   peer address ipv4 133.150.100.2
!
```

3.7.3.4 Identical Policies on all KSs

With multiple COOP KSs, the policies configured on each KS must be considered. **It is recommended that the same GET VPN policies should be configured on each of the KSs.** If a different COOP KS assumes the primary role, it should distribute the same rules in a rekey message to the GMs. If the policies were different, the GMs would receive different policies whenever a different KS is elected as primary. This can cause disruption.

To minimize the time that inconsistent policies exist on KSs when policy changes are required, the order of change to the policy is important. When policy is changed on a primary KS, the new policy is immediately distributed to the GMs for execution. If a secondary KS has the old policy, any GM registering with the secondary KS receives the old policy. However, the GMs receive the new policy during the rekey process that occurs at the expiration of the TEKs associated with the new policies (by default, TEKs expire after an hour).

Therefore, it is recommended to first make the new policy change on all secondary KSs. Subsequently, a policy change on the primary KS forces all GMs to use the new policy as a result of a rekey. If a GM registers before the policy change is executed on the primary KS, that GM will use the new policy as defined on a secondary KS. However, if a rekey occurs before making the changes to the primary KS policy, the old policy would be applied to all GMs even if they received the new policy during registration

Tip: To minimize the potential policy discrepancies, policy changes should be applied near the end of the TEK lifetime, but before rekey).

See 2.2.4 "Configuring Access List Policies" for more information about configuring KS policies.

3.7.3.5 RSA Key on COOP KSs

KSs require RSA keys to create KEK keys. In GET VPN, RSA keys are generated only on KSs and are used to authenticate and sign rekey messages. The KS creates a RSA public and private key pair. The public key is downloaded to all GMs at registration. The KS signs the rekeys with the private key and all GMs verify the rekey messages using the public key.

The RSA key pair must be identical on all KSs. If a KS is added to the group without the RSA key, the new KS cannot create policies. The new KS can still register GMs, but without policies, GMs stay in a fail-closed mode with no lifetime expiration. Any GMs that registered to the new KS would have to be cleared using **`clear crypto gdoi`** and then register to a KS with a valid RSA key.

Tip: If a KS supports multiple groups, a unique RSA key pair can be established for each group.

The RSA key must be synchronized to all KSs. The easiest method is to generate the RSA key on the first KS and make the RSA exportable. Then, save the RSA key (public and private) on a secure backend system. As KSs are added, they can import the key.

See 2.4.1 "Exporting and Importing RSA Keys" for configuration information for importing and exporting RSA keys.

Note: When using PKI, it is recommended to create a separate RSA key for PKI purposes. This enables modification of the PKI keys without affecting COOP between the KS. This is also true for modification of the RSA key used for COOP on the KS.

3.7.3.6 Network Convergence before COOP Election

If routing or network convergence takes place before KS COOP peer detection, routing convergence minimally impacts COOP. However, if network convergence takes longer, some KSs cannot reach other KSs during COOP election. This might cause multiple KSs to be elected as primary (see 3.7.4 Failure Conditions for more details). Multiple primary KSs creates an artificial network partition and can lead to multiple rekeys being sent in a short interval after the network converges.

During regular operation of the COOP KSs, the KS COOP peer detection process should take longer than the expected routing and network convergence interval. Failure to make the KS COOP DPD interval longer than the routing convergence interval induces network instability by forcing a KS network partition, followed by a KS network merge.

3.7.3.7 Buffer Size Configuration

With large number of GMs (1000+) and large policy statements, COOP and rekey message sizes can grow quite large (see 3.7.1.1 “COOP ” for COOP message sizes). Small buffers prevent messages from being transmitted efficiently and increase the potential for failed transmission of announcement and rekey messages. It is recommended to increase the HUGE buffer to its maximum value.

To increase the HUGE buffer using CLI, use the following:

```
!
buffers huge size 65535
!
```

3.7.3.8 Backup Link between KSs

During a network split, COOP KSs may lose connectivity to each other. This might lead to multiple KS operating in primary mode. This results in GMs in different portions of the split network having different keys. While the GMs continue to operate, there are cases when GMs have complete connectivity, but KSs can experience a network split that can lead to loss of communication between GMs. Whenever KSs lose connectivity with the primary KS, multiple rekeys might be exchanged in the system as new primaries are elected. This can be quite disruptive.

To increase resiliency, it is highly recommended to provide multiple paths between the COOP KSs, such as with an out of band network backup. This path should not be inline with the data plane, and preferably a separate link. This kind of a backup link provides a continuous channel between the COOP KSs, ensures that they remain synchronized, prevents fluctuation in primary roles, and prevents unnecessary rekeys being sent.

The disadvantage of using a backup link is that during the network split, certain GMs might not be able to reach the primary KS (see 3.7.4.2 “Network Split and Merge” for more information about how this impacts GMs). This causes GMs to re-register with the next available KS, and this process continues if the network failure is considerably long. As the number of GMs in the system increase, a large number of re-registrations might be seen periodically.

To prevent this scenario, the backup link should facilitate rekey message distribution. This is easily accomplished using unicast rekey by ensuring that the GM IP identities are reachable through the backup link. Similarly, KS IP identities must be reachable to the GM via the backup link.

If multicast is used, the multicast rekey must pass through the backup link. The method used to forward the multicast rekeys on the backup link varies according the multicast architecture (PIM-Sparse Mode (PIM-SM), PIM-Source Specific Mode (PIM-SSM), PIM-SM Anycast, and Multicast Source Discovery Protocol (MSDP)). PIM must operate on the backup link in most cases.

Failure to facilitate rekeys via the backup link results in persistent re-registration by GM that cannot receive the rekey messages.

3.7.4 Failure Conditions

This section explains some common failure scenarios that can be seen with COOP KSs. As previously explained, the primary KS is responsible for sending out the periodic rekeys while the secondary KSs maintain complete state information for GMs and are ready to take over if the primary KS fails. If a secondary KS takes over the primary role, the new primary KS sends rekeys to all GMs. If there are multiple transitions of primary role among COOP KSs, there can be many rekeys sent out in the network, which can be very disruptive. Multiple rekeys also increases IPsec SAs at the GMs, which can lead to memory issues in some GMs. When there are many rekeys in a network, it is important to find the cause and resolve it immediately.

Secondary KSs can lose connectivity with the primary KS for the following reasons:

- KS failure – If the primary KS fails, a secondary KS takes over.
- Network partitioning – A network split can lead to multiple KSs losing connectivity to the primary KS or other KSs. This can lead to multiple KSs declaring themselves as primary KSs. While this provides for redundancy, it can lead to problems from a network split, or from GMs that connect to multiple primary KSs.

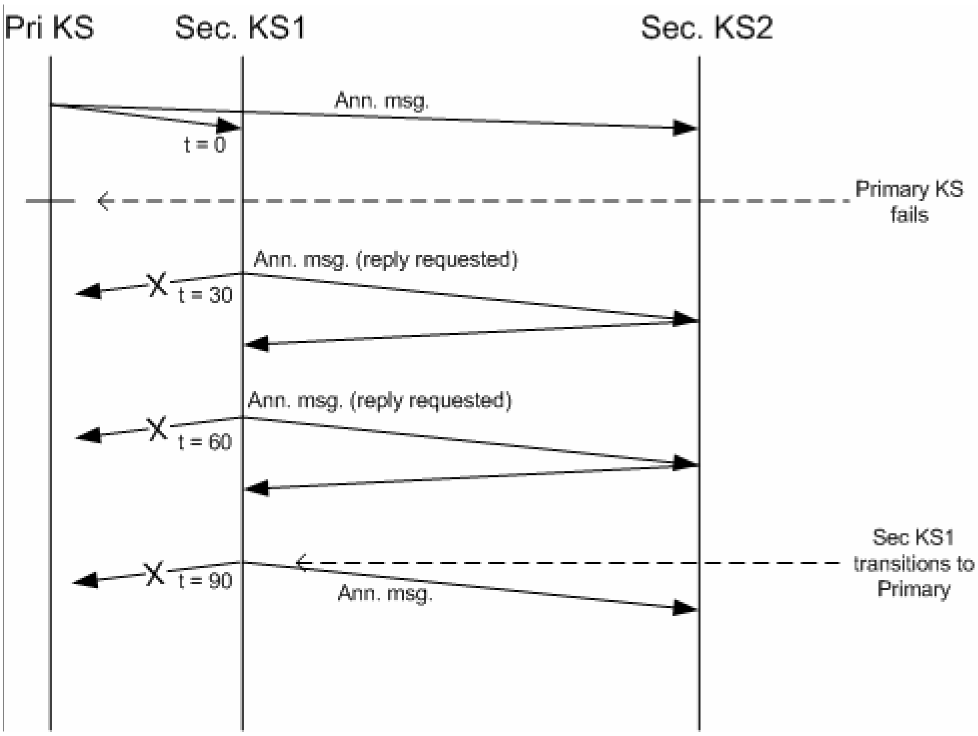
3.7.4.1 KS Failure

When Primary KS fails:

The primary KS sends an announcement message every 20 seconds (the default policy update timer period). If this message is not received by a secondary KS, the secondary KS sends announcement messages to all peer KSs requesting a response. If there is no reply from the primary KS, the secondary KS with the highest priority takes over as the primary KS. From the time the last announcement message is received from the failing primary KS, it takes 90 seconds for a new KS to transition to primary state.

Figure 3-9 shows secondary KS behavior just before it transitions to the primary state.

Figure 3-9: COOP KS: Secondary to Primary Transition



The following IOS snippets are from Secondary KS1.

Initially, Sec-KS1 shows Pri-KS as active and Sec-KS2 as unknown

```
Sec-KS1#show crypto gdoi ks coop
<..>
        Local Address: 139.1.1.1
        Local Priority: 100
        Local KS Role: Secondary , Local KS Status: Alive
<..>

        Peer Sessions:
        Session 1:
                Server handle: 2147483651
                Peer Address: 139.1.1.3
                Peer Priority: 1
                Peer KS Role: Secondary , Peer KS Status: Unknown
<..>

        Session 2:
                Server handle: 2147483656
                Peer Address: 139.1.1.2
                Peer Priority: 175
                Peer KS Role: Primary   , Peer KS Status: Alive
<..>
```

When Pri-KS fails (hardware failure or reboot), Sec-KS1 later transitions to primary state and sends a rekey to the GDOI group. The following syslog messages from Sec-KS1 are seen during this process:


```
Sec-KS1#
May  7 10:21:25.327: %GDOI-5-COOP_KS_TRANS_TO_PRI: KS 139.1.1.1 in group
dgvpn1 transitioned to Primary (Previous Primary = 139.1.1.2)

May  7 10:21:30.328: %GDOI-3-COOP_KS_UNREACH: Cooperative KS 139.1.1.2
Unreachable in group dgvpn1

May  7 10:21:32.628: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey
for group dgvpn1 from address 139.1.1.1 with seq # 1

May  7 10:21:50.329: %GDOI-3-COOP_KS_UNREACH: Cooperative KS 139.1.1.2
Unreachable in group dgvpn1

May  7 10:22:10.330: %GDOI-3-COOP_KS_UNREACH: Cooperative KS 139.1.1.2
Unreachable in group dgvpn1

May  7 10:22:30.331: %GDOI-3-COOP_KS_UNREACH: Cooperative KS 139.1.1.2
Unreachable in group dgvpn1
```

The **COOP_KS_TRANS_TO_PRI** message indicates a change in primary KS status: KS 139.1.1.1 is the new primary KS; the previous primary KS was KS 139.1.1.2.

The **KS_SEND_UNICAST_REKEY** shows that this KS has sent out a rekey message to the group dgvpn1, with rekey seq# 1.

The **COOP_KS_UNREACH** message indicates that this KS can no longer reach KS 139.1.1.2. Note that these messages are 20 seconds apart (the policy update timer period).

After transitioning to primary, Sec-KS1 now shows Sec-KS2 (139.1.1.3) as Secondary/Alive and shows Pri-KS as Primary/Dead.

```
Sec-KS1#show crypto gdoi ks coop

<..>
        Local Address: 139.1.1.1
        Local Priority: 100
        Local KS Role: Primary   , Local KS Status: Alive
<..>

        Peer Sessions:
        Session 1:
                Server handle: 2147483651
                Peer Address: 139.1.1.3
                Peer Priority: 50
                Peer KS Role: Secondary , Peer KS Status: Alive
<..>

        Session 2:
                Server handle: 2147483656
                Peer Address: 139.1.1.2
                Peer Priority: 1
                Peer KS Role: Primary   , Peer KS Status: Dead
                Antireplay Sequence Number: 11063

<..>
```


3.7.4.1.1 Role of the New Primary KS

When a KS transitions to primary state, it does the following:

- Sends an announcement message to all KSs. Other active KS learn of the new primary from this message.
- Sends out rekeys with a new TEK to all GMs.

3.7.4.1.2 When the Previous Primary KS Recovers:

When the previous primary KS recovers, it sends an announcement message to all KSs listed in its configuration. Depending on the response, it may or may not become a primary KS. The possible scenarios are:

- The recovering KS receives an announcement message reply from an existing primary, which has lower priority. In this case, there is no preemption, and the recovering KS remains a secondary KS. This eliminates unnecessary changes in the system.
- The recovering KS receives an announcement message reply from an existing primary, which has higher priority. In this case, the recovering KS remains a secondary KS.
- The recovering KS does not receive a response from a primary KS. In this case, if the recovering KS has the highest priority when compared to other KSs that responded, the recovering KS transitions to primary state.

For a KS that recovers after a reboot or power cycle, network convergence is an important consideration. If network convergence takes longer than the COOP process, the KS cannot see any messages from other KSs. This emulates a network split (see 3.7.4.2.3 Network Split and Merge (KS Split)), even though there is no real network split. This might cause unnecessary rekeys to be transmitted after the network converges. It is preferable for network convergence to occur faster than the COOP process at KS initialization.

Continuing the example in Figure 3-9, when Pri-KS recovers, it initially comes up in secondary state. After the network converges (COOP process is not completed yet), Pri-KS detects Sec-KS1 as a primary; Pri-KS retains its secondary state. Sec-KS1 updates Pri-KS now as Secondary/Alive.

The following message is seen at Sec-KS1 during this process.

The message **`COOP_KS_REACH`** shows that Sec-KS1 has its connectivity with KS 139.1.1.2 (Pri-KS) restored.

```
Sec-KS1#
May  7 10:51:30.473: %GDOI-5-COOP_KS_REACH: Reachability restored with
Cooperative KS 139.1.1.2 in group dgvpn1.
```

3.7.4.2 Network Split and Merge

Network instability can lead to loss of connectivity between the primary KS and the secondary KSs. This is referred to as a network split in the context of the COOP KS. The network split might last for a few seconds, or through multiple rekey intervals. Depending on the interval, you might see slightly different behaviors.

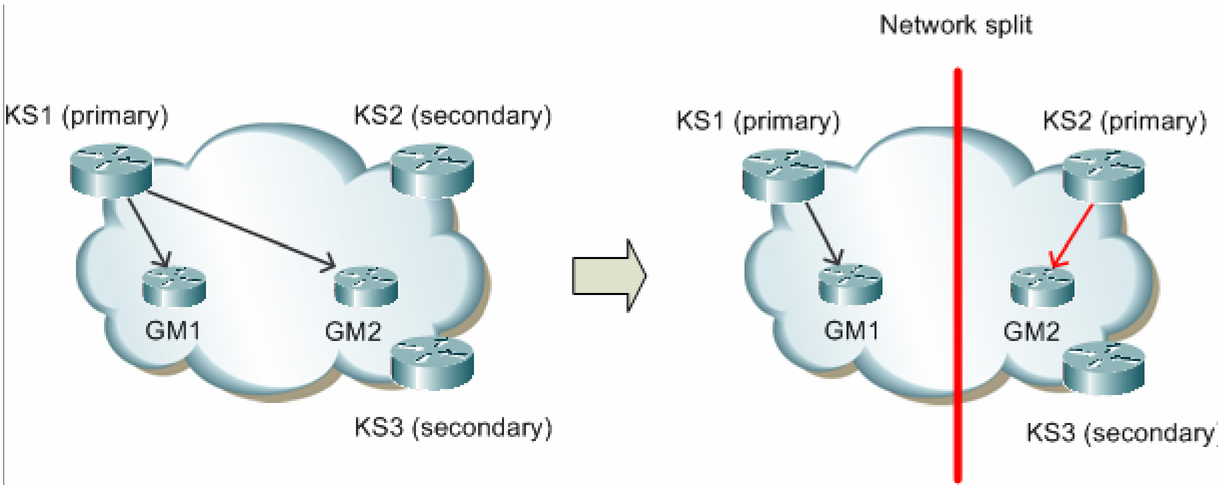
3.7.4.2.1 Network Split and Merge (KS and GM Split)

Consider a scenario where a customer purchased MPLS VPN services from two SPs who use Inter-AS services to join the two regions. If the Inter-AS link fails and there are KSs on both sides of the Inter-AS link, GMs will obtain key material from the KS in their respective partition.

Network Split

Figure 3-10 shows that initially KS1 is the primary KS and provides the keys for GMs GM1 and GM2. After a network split, KS2 also becomes a primary KS. Now, GM1 receives its key from KS1, while GM2 receives its key from KS2.

Figure 3-10: Network Split and Merge: KS and GM Split



The sequence of events follows:

1. Initially, KS1 and KS2 have connectivity and KS1 (the primary KS) sends the TEK in the rekey messages to GM1 and GM2.
2. A network split occurs that isolates KS1 and GM1 from KS2, KS3, and GM2.
3. KS2 and KS3 detect the loss of connectivity with KS1 and KS2 transitions to primary state.
4. KS2 issues a rekey to the network and provides a new KEK and TEK to the GMs. The rekey message from KS2 contains the old TEK and the new TEK. GM2 continues to use the old TEK for encryption because it has a lifetime which expires sooner.
5. In the case of a unicast rekey, GM2 responds and KS2 eventually times out GM1. In the case of a multicast rekey, KS2 is not aware of GM1s state.
6. On TEK-rollover (rekey interval), KS1 issues a new TEK. In the case of a unicast rekey, KS1 eventually times out GM2. In case of a multicast rekey, KS1 is not aware of GM2's state.
7. Now, GM1 has a different TEK and KEK as compared to GM2.

The preceding network split scenario is explained further using CLI outputs from KS1, KS2, GM1, and GM2.

1. Initial state: The following show commands display KS status and roles and TEK and KEK values at the KSs and GMs.

   Initially, KS1 shows KS2 and KS3 as Secondary/Alive, while KS2 shows KS1 as Primary/Alive.

```
!KS1
KS1#show cry gdoi ks coop
<..>
        Local Address: 139.1.1.2
        Local Priority: 175
```



```
        Local KS Role: Primary   , Local KS Status: Alive
<..>
                Peer Address: 139.1.1.3         <-- KS3 status
                Peer Priority: 50
                Peer KS Role: Secondary , Peer KS Status: Alive
<..>
                Peer Address: 139.1.1.1         <-- KS2 status
                Peer Priority: 100
                Peer KS Role: Secondary , Peer KS Status: Alive

!KS2
KS2#sh cry gdoi ks coop
<..>
        Local Address: 139.1.1.1
        Local Priority: 100
        Local KS Role: Secondary , Local KS Status: Alive
<..>
                Peer Address: 139.1.1.3         <-- KS3 status
                Peer Priority: 50
                Peer KS Role: Secondary , Peer KS Status: Unknown
<..>
                Peer Address: 139.1.1.2         <-- KS1 status
                Peer Priority: 175
                Peer KS Role: Primary   , Peer KS Status: Alive
```

The initial TEK and KEK value at KS1 and KS2 are the same. The following shows only the value at KS1:

```
!KS1
KS1#show cry gdoi ks policy
<..>
  KEK POLICY (transport type : Unicast)
    spi : 0x8D35D4A8241E2A34888E3D1AA6D1F9FA
<..>

  TEK POLICY (encaps : ENCAPS_TUNNEL)
    spi               : 0x2D45C4E9    access-list           : 160
<..>
```

Initially, GM1 and GM2 have the same TEK value, as seen in the active security parameter index (SPI):

```
!GM1
GM1#show cry ipsec sa
<..>
     inbound esp sas:
      spi: 0x2D45C4E9(759547113)
        Status: ACTIVE
<..>
     outbound esp sas:
      spi: 0x2D45C4E9(759547113)
        Status: ACTIVE

!GM2
GM2#show cry ipsec sa
<..>
     inbound esp sas:
      spi: 0x2D45C4E9(759547113)
        Status: ACTIVE
```



```
<..>
     outbound esp sas:
      spi: 0x2D45C4E9(759547113)
        Status: ACTIVE
```

Initially, both GM1 and GM2 have the same KEK:

```
!GM1
GM1#show cry gdoi gm rekey
<..>
Rekey (KEK) SA information :
          dst             src             conn-id  my-cookie  his-cookie
New     : 144.150.100.1   139.1.1.1        13039   888E3D1A   8D35D4A8
Current : ---             ---               ---    ---        ---
Previous: ---             ---               ---    ---        ---


!GM2
GM2#show cry gdoi gm rekey
<..>
Rekey (KEK) SA information :
          dst             src             conn-id  my-cookie  his-cookie
New     : 138.1.1.1       139.1.1.1        13638   888E3D1A   8D35D4A8
Current : ---             ---               ---    ---        ---
Previous: ---             ---               ---    ---        ---
```

2. The network split occurs. The following CLI snippets are from the CLI of the KSs and GMs, showing KS statuses. The TEK and KEK values at the GMs are also compared.

   At the network split, KS2 becomes the new primary. Here are the log messages from KS2:

```
!message to indicate change in primary
May  8 09:39:17.960: %GDOI-5-COOP_KS_TRANS_TO_PRI: KS 139.1.1.1 in group
dgvpn1 transitioned to Primary (Previous Primary = 139.1.1.2)

!message indicates that when this KS transitions to primary, it sends out
a rekey with new key values to known GMs
May  8 09:39:17.992: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey
for group dgvpn1 from address 139.1.1.1 with seq # 1

!message indicating that KS2 can't reach KS1
May  8 09:39:22.960: %GDOI-3-COOP_KS_UNREACH: Cooperative KS 139.1.1.2
Unreachable in group dgvpn1
```

After the split, KS1 shows KS2 and KS3 as dead, and KS2 also shows KS1 as dead.

```
!KS1
KS1#show cry gdoi ks coop
<..>
        Local Address: 139.1.1.2
        Local Priority: 175
        Local KS Role: Primary   , Local KS Status: Alive
<..>
                Peer Address: 139.1.1.3         <-- KS3 status
                Peer Priority: 50
                Peer KS Role: Secondary , Peer KS Status: Dead
```



```
<..>
                Peer Address: 139.1.1.1         <-- KS2 status
                Peer Priority: 100
                Peer KS Role: Secondary , Peer KS Status: Dead

!KS2
KS2#show cry gdoi ks coop
<..>
        Local Address: 139.1.1.1
        Local Priority: 100
        Local KS Role: Primary   , Local KS Status: Alive
<..>
                Peer Address: 139.1.1.3         <-- KS3 status
                Peer Priority: 50
                Peer KS Role: Secondary , Peer KS Status: Alive
<..>
                Peer Address: 139.1.1.2         <-- KS1 status
                Peer Priority: 1
                Peer KS Role: Primary   , Peer KS Status: Dead
```

After the split, KS2 sends out a rekey.

The following log message is from GM2, indicating that GM2 received a rekey from KS2.

```
!message indicates that GM2 (138.1.1.1) received rekey from
!KS2(139.1.1.1). Rekey sequence number is 1

May  8 10:39:18.054: %GDOI-5-GM_RECV_REKEY: Received Rekey for group dgvpn
from 139.1.1.1 to 138.1.1.1 with seq # 1
```

After the split, KS2 has a different TEK and KEK when compared to KS1.

```
!KS2
KS2#show cry gdoi ks policy
<..>
  KEK POLICY (transport type : Unicast)
    spi : 0x78BA208E3A8BC08A471AC7A787950926
<..>
  TEK POLICY (encaps : ENCAPS_TUNNEL)
    spi                : 0x30808D3D    access-list           : 160
```

After the split, GM2 receives a rekey from KS2, so it now has a different TEK when compared to GM1. GM2 should have the old and the new TEK.

```
!GM2
GM2#show cry ipsec sa
<..>
     inbound esp sas:
      spi: 0x30808D3D(813731133)
        Status: ACTIVE
<..>
     outbound esp sas:
      spi: 0x30808D3D(813731133)
        Status: ACTIVE
```


After three rekeys, KS1 eventually times out GM2 (This is only in case of unicast rekey, as used in this example). The GM2 state at KS1, after the third rekey and its retransmits are attempted, is as follows:

```
!KS1
KS1#sh cry gdoi ks mem | b 138.1.1.1      <-- check GM2 status
Group Member ID    : 138.1.1.1
Group ID           : 61440
Group Name         : dgvpn1
Key Server ID      : 139.1.1.1
Rekeys sent        : 3
Rekeys retries     : 6
Rekey Acks Rcvd    : 0
Rekey Acks missed  : 2

Sent seq num :     7    8    9    0       <-- seq#1 was the 1st rekey, &
Rcvd seq num :     0    0    0    0             seq#2 & 3 are retransmits
                                              seq #4 is the 2nd rekey
                                              seq #7 is the 3rd rekey &
                                                seq#8 & 9 are retransmits
                                              For this example, rekey was
                                                configured 30 sec retrans-
                                                mit interval, & 2 retrans-
                                                mit count
```

After the third rekey, that is, just before the fourth rekey is sent, KS1 deletes GM2.

Similarly, KS2 deletes GM1.

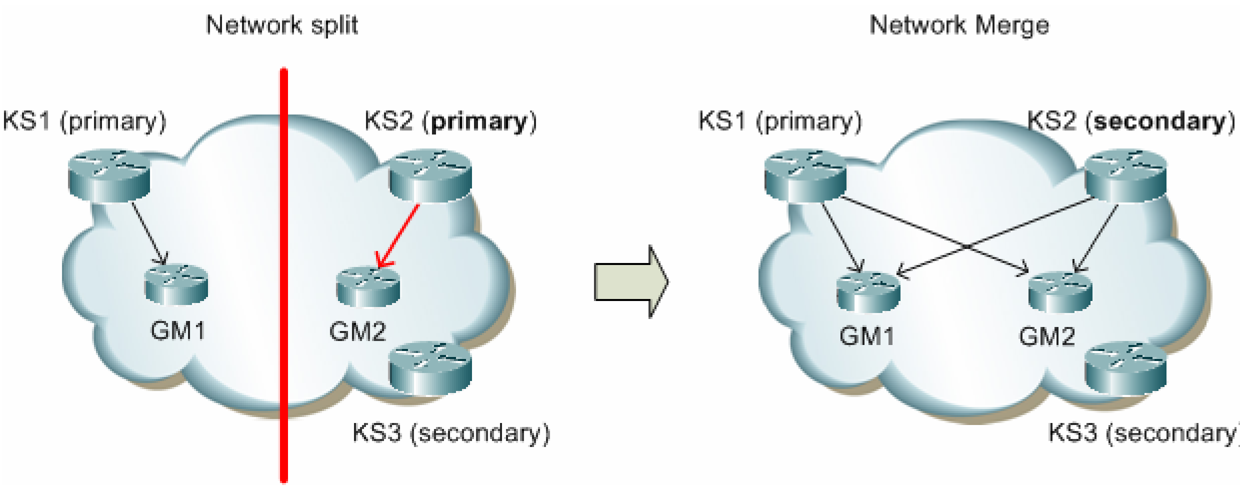
The following log messages from KS1 and KS2 indicate that the GM is deleted.

```
!From KS1
May  8 11:30:28.026: %GDOI-4-GM_DELETE: GM 138.1.1.1 deleted from group
dgvpn1.


!From KS2
May  8 11:30:08.157: %GDOI-4-GM_DELETE: GM 134.150.100.1 deleted from
group dgvpn1.
```

3.7.4.2.2 Network Merge

If the network split is long enough, GM1 and GM2 will have a different KEK and TEK. So, when a merge occurs, the primary KS must ensure that the GMs understand the new rekey information. Therefore, the primary KS sends out rekey information that is encrypted with both KEKs, as shown in Figure 3-11.

Figure 3-11: Network Split and Merge: KS and GM Merge



The sequence of events follows:

1. Before the merge, GM1 and GM2 have a different KEK, and might have a different TEK. (This depends on the duration of the network split. If the split lasted longer than the TEK rollover, the GMs will have different TEKs.)
2. After the merge, KS1 and KS2 have their connectivity restored. KS2 discovers KS1 is in primary state and has higher priority, so KS2 demotes itself to secondary state.
3. KS1 now issues a rekey, encrypted using the existing KEK. KS2 issues a rekey with its old KEK and informs all GMs to now use the same KEK as KS1. Both the rekeys from KS1 and KS2 also include the multiple TEKs.
4. After the rekey, GM2 reverts to using the same KEK as GM1.
5. After the rekey, GM1 and GM2 have both TEK values. They continue to use the TEK that has the shortest remaining lifetime.

This network merge scenario is explained further using CLI outputs from KS1, KS2, GM1, and GM2.

```
!messages from KS1
May  8 11:36:38.066: %GDOI-5-COOP_KS_REACH: Reachability restored with
Cooperative KS 139.1.1.3 in group dgvpn1.

May  8 11:36:48.066: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey
for group dgvpn1 from address 139.1.1.2 with seq # 11


!messages from KS2
May  8 10:36:38.213: %GDOI-5-COOP_KS_REACH: Reachability restored with
Cooperative KS 139.1.1.2 in group dgvpn1.

May  8 10:36:48.033: %GDOI-5-COOP_KS_TRANS_TO_PRI: KS 139.1.1.2 in group
dgvpn1 transitioned to Primary (Previous Primary = 139.1.1.1)

May  8 10:36:48.065: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey
for group dgvpn1 from address 139.1.1.1 with seq # 11
```


KS1 shows KS2 as Secondary/Alive, while KS2 shows KS1 as Primary/Alive. CLI output from KS2 follows:

```
!KS2
KS2#show cry gdoi ks coop
<..>
        Local Address: 139.1.1.1
        Local Priority: 100
        Local KS Role: Secondary , Local KS Status: Alive
<..>
                Peer Address: 139.1.1.3          <-- KS3 status
                Peer Priority: 50
                Peer KS Role: Secondary , Peer KS Status: Unknown
<..>
                Peer Address: 139.1.1.2          <-- KS1 status
                Peer Priority: 175
                Peer KS Role: Primary   , Peer KS Status: Alive
```

The TEK and KEK values at KS1 are shown. Similar keys can be seen at KS2 (not shown here)

Note that KS1 has two sets of keys: one under server 139.1.1.2 and another under server 139.1.1.1 (KS2). The KEK value is the same for both sets, but they have different TEK values.

```
!KS1
KS1# show cry gdoi ks policy

For group dgvpn1 (handle: 2147483650) server 139.1.1.2 (handle:
2147483650):

  # of teks : 1  Seq num : 11
  KEK POLICY (transport type : Unicast)
    spi : 0x8D35D4A8241E2A34888E3D1AA6D1F9FA
<..>
  TEK POLICY (encaps : ENCAPS_TUNNEL)
    spi                : 0x5B35AD05    access-list           : 160
<..>

For group dgvpn1 (handle: 2147483650) server 139.1.1.1 (handle:
2147483653):

  # of teks : 1  Seq num : 0
  KEK POLICY (transport type : Unicast)
    spi : 0x8D35D4A8241E2A34888E3D1AA6D1F9FA
<..>
  TEK POLICY (encaps : ENCAPS_TUNNEL)
    spi                : 0x8383AE97    access-list           : 160
<..>
```

Now we look at GM states. After the merge, both KSs sent out rekeys. Log messages from GM1 and GM2 indicate that both GMs received rekeys from each KS.

```
!GM1

!rekey received from KS1 (139.1.1.2)
May  8 11:36:48.120: %GDOI-5-GM_RECV_REKEY: Received Rekey for group dgvpn
from 139.1.1.2 to 134.150.100.1 with seq # 11
```



```
!GM2
!rekey received from KS2 (139.1.1.1)
May  8 11:36:48.131: %GDOI-5-GM_RECV_REKEY: Received Rekey for group dgvpn
from 139.1.1.1 to 138.1.1.1 with seq # 11

!rekey received from KS1 (139.1.1.2)
May  8 11:36:48.211: %GDOI-5-GM_RECV_REKEY: Received Rekey for group dgvpn
from 139.1.1.2 to 138.1.1.1 with seq # 11
```

Each GM now has both TEKs and uses the TEK with the shortest remaining lifetime.

```
!GM1
GM1#show cry ipsec sa

<..>
     inbound esp sas:
      spi: 0x5B35AD05(1530244357)
        Status: ACTIVE
      spi: 0x8383AE97(2206445207)
        transform: esp-aes esp-sha-hmac ,
        Status: ACTIVE
<..>
     outbound esp sas:
      spi: 0x5B35AD05(1530244357)
        Status: ACTIVE
      spi: 0x8383AE97(2206445207)
        Status: ACTIVE

!GM2
GM2#show cry ipsec sa
<..>
     current outbound spi: 0x8383AE97(2206445207)

     inbound esp sas:
      spi: 0x8383AE97(2206445207)
        Status: ACTIVE
      spi: 0x5B35AD05(1530244357)
        Status: ACTIVE
<..>
      spi: 0x8383AE97(2206445207)
        Status: ACTIVE
      spi: 0x5B35AD05(1530244357)
        Status: ACTIVE
```

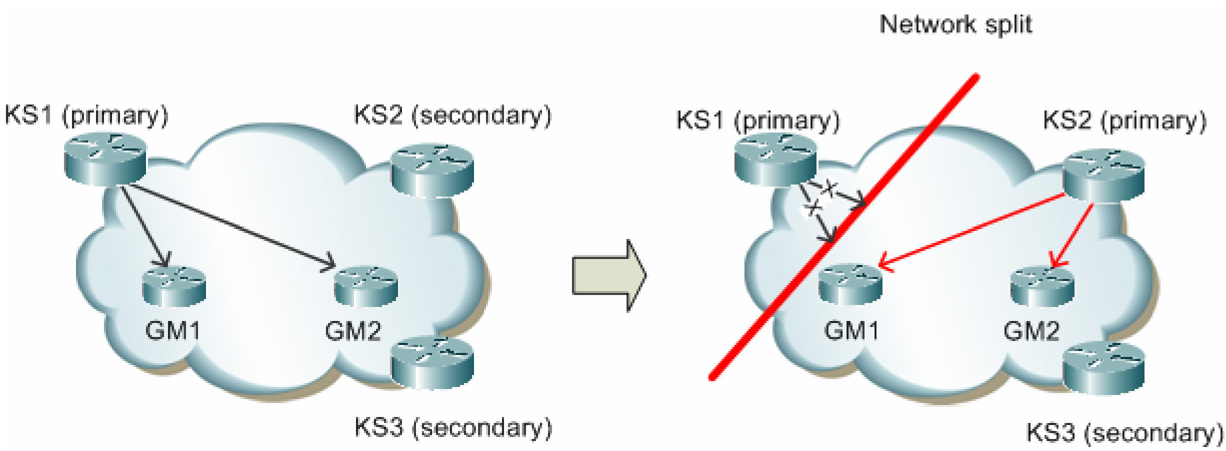
To summarize: During this kind of network split, GMs have connectivity to the KSs in their respective partition, but do not have connectivity across partitions. After the network split is resolved, the primary KS sends out multiple rekeys, each encrypted using one of the different KEKs, so that all GMs understand this rekey information and synchronize to the same set of keys.

3.7.4.2.3 Network Split and Merge (KS Split)

A network split can occur so that only the primary KS is isolated from rest of the network. When KS2 does not receive any update messages from KS1, KS2 takes over as the primary role and sends out a rekey to the GMs with a new KEK and TEK.

As shown in Figure 3-12, in this scenario, rekeys sent by the isolated primary KS are not received by any GMs, and eventually time out on all GMs. KS2 assumes the responsibility of primary KS and rekeys all GMs. If any GMs fail to receive the rekey, they re-register with KS2.

Figure 3-12: Network Split and Merge: Only KS Split



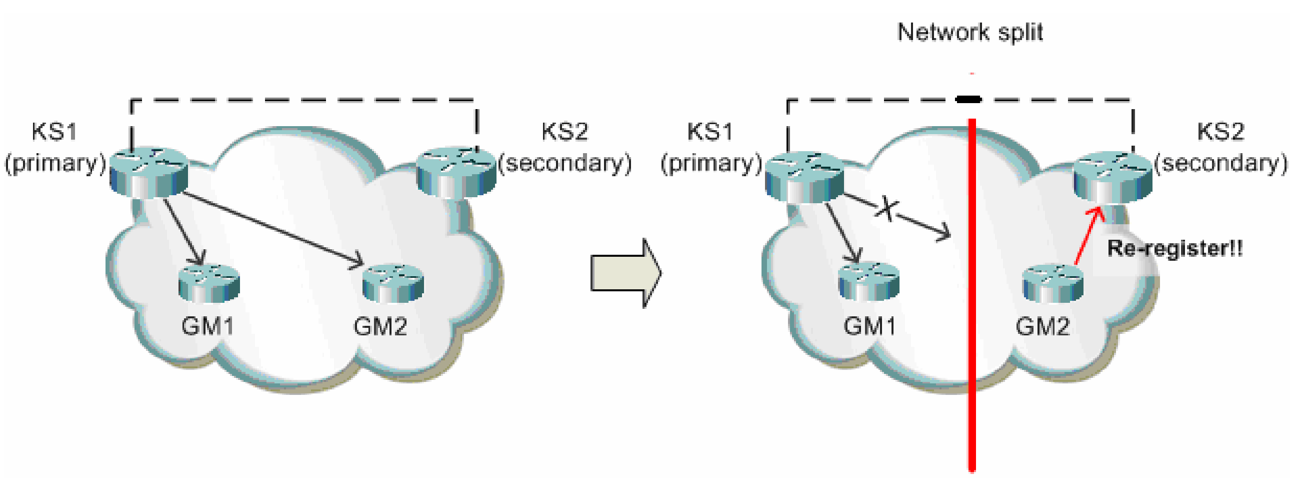
This scenario is almost the same as that described in 3.7.4.1 "KS Failure." The main difference in this scenario is that during the split, KS1 and KS2 both become primary KSs, but that rekeys from KS1 never makes it to any GM. When the network merge occurs, KS2 sees another higher priority KS, and so rolls back to secondary state.

3.7.4.2.4 Network Split and Merge (GM Split, KSs Connected)

In cases that employ KS-to-KS backup link, a network split may occur such that GMs have reachability to the KS in their partition, but KSs retain full connectivity to each other. In this case, the GM that is unable to reach the primary KS will not be able to receive a rekey message and will be forced to re-register with the secondary KS. The KS will still synchronize their database such that the GM will continue to receive a consistent set of KEK and TEK. This example highlights the importance of providing an alternative path between the KS in order to maintain the integrity and consistency of the GM database and key sets.

Figure 3-13 shows a network split, in which the KSs have full connectivity and are able to exchange state information. However, GM2 is unable to reach the primary KS (KS1).

Figure 3-13: Network Split & Merge: only GM split



If GM2 is configured with both KSs in its configuration, it will eventually re-register with KS2. KS2 will be able to inform KS1 about GM2's state. KS1 will attempt to issue rekey to GM2 (in case of unicast rekey), but GM2 will not receive any rekey and will always end up re-registering with KS2.

In summary, if KSs have complete connectivity during a network split, there might be some GMs that will not be able to receive rekey information during the interval of the split. If those GMs have all KSs configured, they will be able to re-register with a secondary KS and obtain valid key information.

Note: Rekey messages (all control plane traffic) could also traverse the backup link between the KS such that all GM receive rekey messages despite the core network partition.

3.8 Designing Around MTU Issues

Because of additional IPsec overhead added to each packet, MTU related issues are very common in IPsec deployments, and MTU size becomes a very important design consideration. If MTU value is not carefully selected by either predefining the MTU value on the end hosts or by dynamically setting it using PMTU discovery, the network performance will be impacted because of fragmentation and reassembly. In the worst case, the user applications will not work because network devices might not be able to handle the large packets and are unable to fragment them because of the df-bit setting. Some of the scenarios which can adversely affect traffic in a GET VPN environment and applicable mitigation techniques are discussed below.

LAN MTU of 1500 – WAN MTU 44xx (MPLS)

In this scenario, even after adding the 50-60 byte overhead, MTU size is much less than the MTU of the WAN. The MTU does not affect GET VPN traffic in any shape or form.

LAN MTU of 1500 – WAN MTU 1500

In this scenario, when IPsec overhead is added to the maximum packet size the LAN can handle (i.e. 1500 bytes) the resulting packet size becomes greater than the MTU of the WAN. The following techniques could help reduce the MTU size to a value that the WAN infrastructure can actually handle.

Manually setting a lower MTU on the hosts

By manually setting the host MTU to 1400 bytes, IP packets coming in on the LAN segment will always have 100 extra bytes for encryption overhead. This is the easiest solution to the MTU issues but is harder to deploy because the MTU needs to be tweaked on all the hosts.

TCP traffic

Configure **`ip tcp adjst-mss 1360`** on GM LAN interface. This command will ensure that resulting IP packet on the LAN segment is less than 1400 bytes thereby providing 100 bytes for any overhead. If the maximum MTU is lowered by other links in the core (e.g. some other type of tunneling such as GRE is used in the core), the adjust-mss value can be lowered further. This value only affects TCP traffic and has no bearing on the UDP traffic.

3.8.1.1 Host compliant with PMTU discovery

For non-TCP traffic, for a 1500 packet with DF bit set, the GM drops the packet and send ICMP message back to sender notifying it to adjust the MTU. If sender and the application is PMTU compliant, this will result in a packet size which can successfully be handled by WAN. For example, if a GM receives a 1500 byte IP packet with the df-bit set and encryption overhead is 60 bytes, GM will notify the sender to reduce the MTU size to 1440 bytes. Sender will comply with the request and the resulting WAN packets will be exactly 1500 bytes.

If the maximum MTU value in the core is lower than 1500 bytes because of additional overheads, on the WAN interface of the GM set the `ip mtu <lowest MTU value in core>`. This will reduce the packet size to what the core can actually handle. For example, if the MTU on a certain link in the core can only be 1400 bytes, set **`ip mtu 1400`** on the WAN interface of GM. When a 1500 byte packet comes in on the LAN segment with df-bit set and the encryption overhead is 60 bytes, the GM will drop the packet and notify the sender to reduce the MTU to 1340 (ip MTU value configured minus the IPsec overhead).

Note. End to end PMTU does not work in GET VPN. Refer to CSCsq23600 for more details

3.8.1.2 Hosts Not Compliant with PMTU Discovery

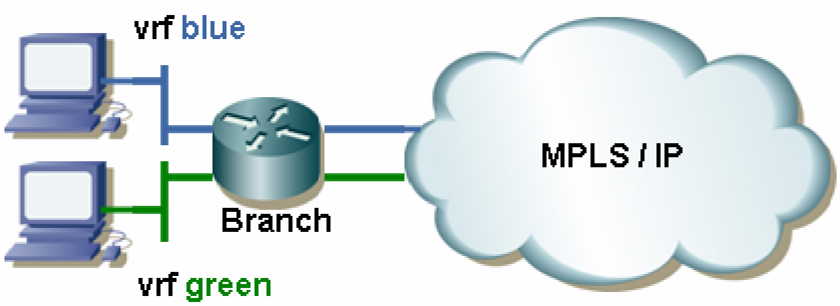
In some cases, end host or the application running on the end host might not be fully compliant with PMTU discovery. This means the host/application is either not setting the df-bit or sets the df-bit but does not reduce the packet size as directed. Additionally, in some cases intermediate routers or firewalls might also drop ICMP messages sent from the GM to the host.

If this traffic is TCP based, tcp adjust-mss will mitigate the problem. If the traffic is not TCP based, as a last resort df-bit can be cleared on the GM using **`crypto ipsec df-bit clear`** in **global** configuration mode. This will result in clearing of the df-bit and packet will be fragmented. Both the IP fragments will then be encrypted. The remote GM will be responsible for decryption while the end host will be responsible for IP packet reassembly. In IOS routers, fragmentation is handled in the CEF path therefore the CPU load does not increase as much due to fragmentation. Reassembly on the other hand is handled in the process path and results in a major CPU impact. Look-ahead fragmentation must be turned on by using **`Crypto ipsec fragmentation before-encryption`** (default in 12.4 releases).

3.9 VRF Aware GET VPN

In certain scenarios, it might be desirable to deploy VRFs in GET VPN solution. It must be noted that GET VPN GM is VRF aware (with some limitations described subsequently) but the GET VPN KS is not. Consider the following deployment:

Figure 3-14 VRF Aware GM



It is possible to configure the VRFs for GET VPN deployment on the GM if the following two criterions are matched:

1. Each VRF will require a unique WAN interface/sub-interface to apply the crypto map
2. Crypto map applied to each VRF will require reference to a unique GET VPN group ID

```
crypto gdoi group blue
identity number 101
server address ipv4 11.1.1.1
!
crypto gdoi group green
identity number 102
server address ipv4 12.1.1.1
!
crypto map blue 10 gdoi
set group blue
!
crypto map green 10 gdoi
set group green
!
interface serial0/0/0:101
 description WAN interface for VRF blue
 ip vrf forwarding blue
 crypto map blue
!
interface Serial0/0/0:1.102
 description WAN interface for VRF green
 ip vrf forwarding green
 crypto map green
!
```

Because the GET VPN KS is currently not VRF aware, GMs should register to a distinct set of KS's per group. This means a KS set per VRF is recommended.

It is possible to cause a set of KS to serve all VRFs by allowing all the GM VRF subinterfaces to connect to a shared interface on the KS. Special security considerations should be taken for such a deployment. Specifically, the KS should use GM authorization and special care must be taken to prevent route leaking. The IKE identity endpoints for all GMs must be routable in a shared routing domain accessible to the KSs. However, the GM IKE identities bound to specific VRFs should not be routable in the other VRFs.

The best practice is to configure the KS set to use a designated route partition specifically for the GDOI management and control plane (for example, network operations center (NOC) routing segment). GM IKE identities can be advertised to the NOC routing segment so that the KS can reach all GM VRF subinterfaces. Meanwhile, the GM IKE identity addresses associated with VRFs are not advertised to other GM VRF routing segments. In fact, the GM IKE identity addresses should be blocked via an ACL on the adjacent PE. This method of route partitioning is commonly used by MPLS VPN providers when managing CE on behalf of a customer.

4 Enterprise Deployment

Depending on size, location, and other parameters, every enterprise has its own network design. Because each customer has a unique set of requirements, there is no single, typical enterprise network layout that can satisfy every design constraint. Although enterprise network designs differ from one customer to another, some common branch and data center (DC) networking elements can be used to select a branch or DC design that meets a particular customer's requirements.

4.1 DC and Branch Designs

In any enterprise network, two major design considerations are the DC (where data is stored securely) and the branches (where data is consumed and generated). The intention of this guide is not to help a user design the DC or branch network, but to help that user seamlessly integrate GET VPN networks into an existing customer deployment.

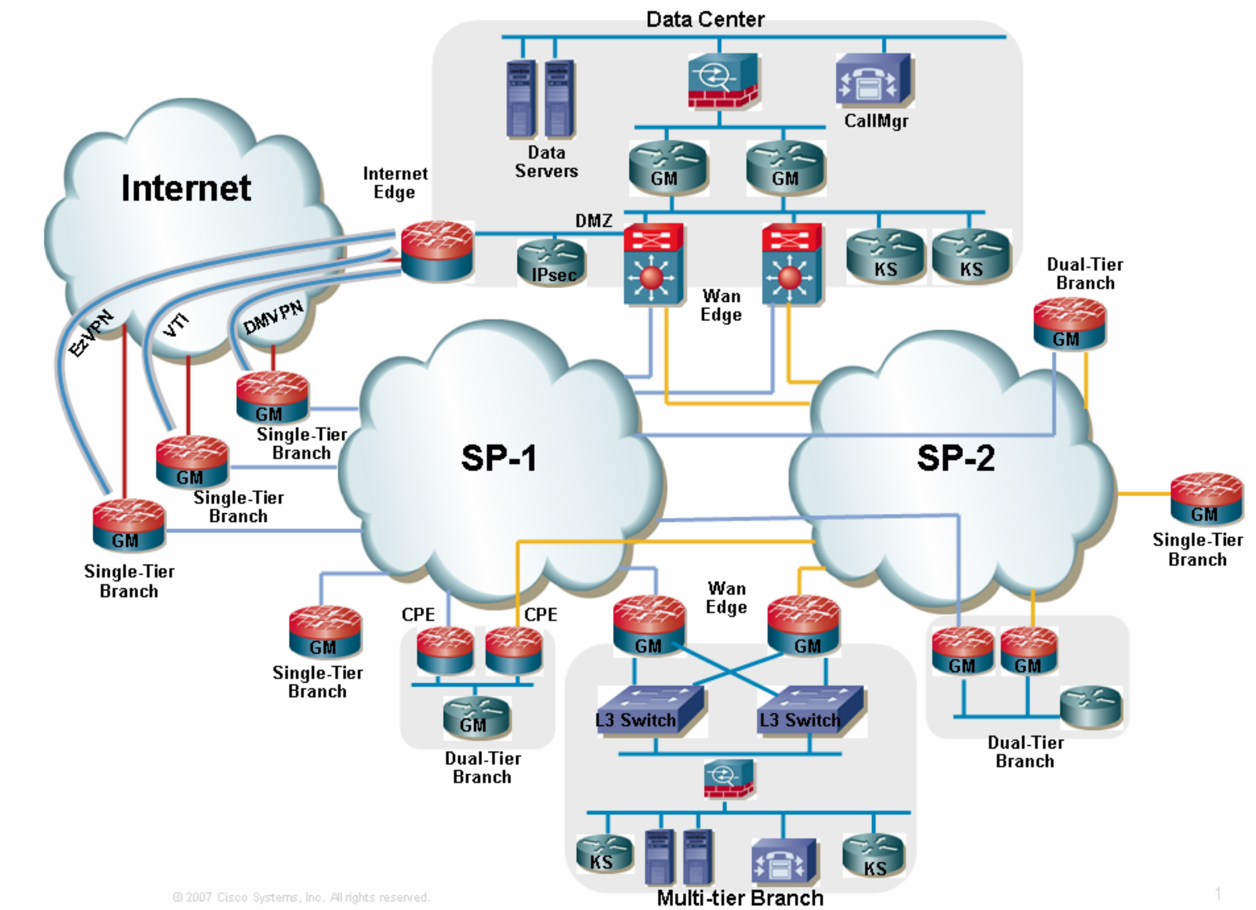
4.1.1 DC Design

This section examines a simplified typical DC network and some of its basic components. The network is then used as an example to explain how GET VPN can be deployed over it. As part of design considerations, the effect of GET VPN on the DC components is explained. We also consider the placement of GET VPN key servers (KSs) and group members (GMs) within the DC, and the corresponding design factors.

4.1.2 Branch Design

The Cisco Enterprise Branch Architecture introduces the concept of three branch profiles that incorporate common branch network components. These three profiles are **not** the only architectures recommended for branch networks, but they represent various aspects of branch network designs. These profiles are used as base lines with which all the integrated services building blocks and application networking services are built. For this guide, a network design was selected that not only follows the Enterprise Branch Architecture recommendation, but also covers certain technology variations within each branch.

Figure 4-1 Network Design Overview



4.1.2.1 Single-Tier Branch Profile

This profile is recommended for smaller enterprise branches that do not require platform redundancy and have a limited number of users. The WAN circuit (T1 or DS3) is terminated on the router (typically ISR), and LAN devices are connected using EtherSwitch module or a Layer 2 (L2) switch. High availability (HA), if required, is achieved through less expensive cable or DSL backup. Because GET VPN does not work over the Internet, a traditional IPsec tunneling solutions is deployed over the backup link.

The following profile variations can be deployed as part of the network design:

- Branch with no backup
- Branch with Dynamic Multipoint VPN (DMVPN) over the backup link
- Branch with EzVPN over the backup link
- Branch with Virtual Tunnel Interface (VTI) over the backup link

4.1.2.2 Dual-Tier Branch Profile

This profile comprises two WAN termination routers, each typically connected to a different WAN provider. Dual WAN links and box redundancy provide a greater level of HA compared to the single-tier branch profile.

This branch is typical of most branches in traditional enterprise branch networks. LAN connectivity is provided by a desktop switch (L2 or L3). Because the GET VPN routers all share the same policies, asymmetric routing is not an issue and we do not need stateful IPsec HA.

Note: If a firewall is enabled on the WAN interfaces, asymmetric traffic flows do not work.

The following profile variations were part of network design:

- Dual WAN termination on one branch router
- Dual WAN termination on dual branch routers
- Dual WAN termination on dual provider customer premises equipment (CPE); GET VPN is terminated on enterprise owned branch router

4.1.2.3 Multitier Branch Profile

This profile consists of dual WAN termination devices, single or dual Adaptive Security Appliance (ASA) appliances for security, dual routers for services integration, and several desktop switches. This profile has the most network gear but provides the greatest amount of HA and redundancy.

The top layer routers provide WAN termination, the ASA appliances provide security services, and the middle layer routers provide integrated services termination. The external desktop switches provide LAN connectivity. GET VPN can be terminated on the WAN aggregation devices (preferred) or services layer. Redundancy and HA are provided at every (or almost every) device. The multitier branch profile closely resembles a small campus and large enterprise branches.

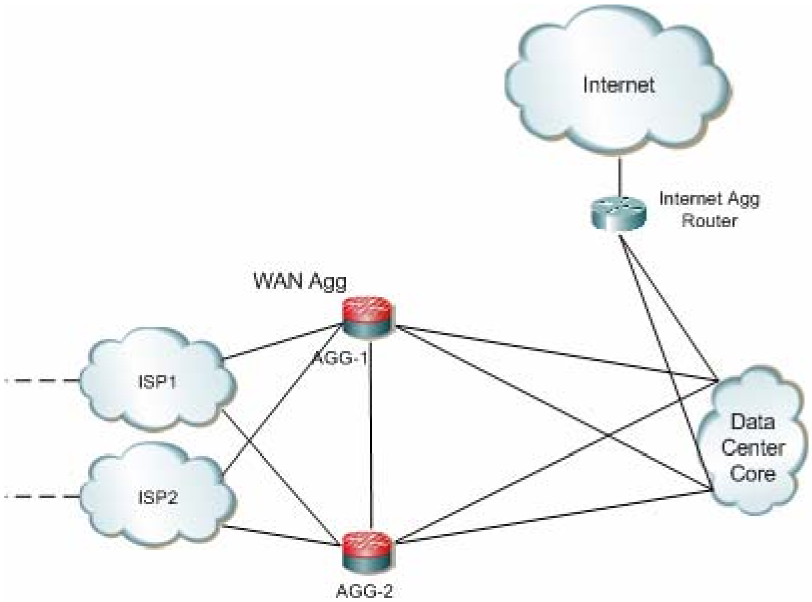
Note: If the GET VPN network is terminated on the WAN termination devices and a KS is also deployed in the branch (as shown in the preceding topology), KS policies must be configured to permit control plane traffic to and from the KS to go through the GM without IPsec encryption.

4.2 DC Design

This section examines GET VPN deployment in a typical DC environment: placement of KSs and GMs and their interaction with existing services such as firewall, quality of service (QoS) and multicast architecture.

Figure 4-2 shows a typical DC design. The WAN aggregation routers AGG-1 and AGG-2 provide MPLS or IP WAN connectivity. Further discussions in this section refer to placement of the KSs and GM in reference to these WAN aggregation devices. DC design is out of scope for this document.

Figure 4-2: DC before GET VPN Deployment



4.2.1 GET VPN DC Design Considerations

. To deploy GET VPN at the DC, consider the following components:

- KS – redundant KSs should be placed at geographically important locations. The DC provides an ideal location to place a KS because it can be administratively monitored with the rest of the DC devices. At the same time, KS can take advantage of the DC Edge router firewall and other security provisions.
- GMs – The traffic to and from the DC needs to be encrypted. One or more GMs provide this functionality.
- Traffic – GET VPN control plane traffic operates on UDP port 848. Consideration needs to be taken to ensure that this traffic can make it to the KSs and GMs as needed.

Tip: For detailed information about KS design, see Chapter 3, "System Design."

4.2.1.1 KS in DC Design

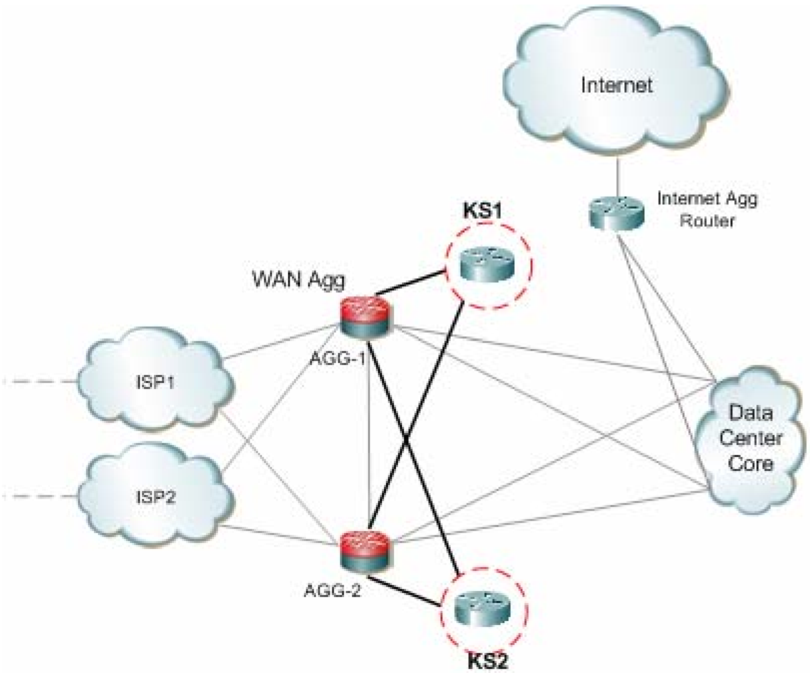
With redundant KSs in the system, KSs can be placed in different locations/sites or they can be put together on a single site. See 3.7.3.1 "Selecting the Number of COOP KSs" and 3.7.3.2 "Setting KS Priority" for recommendations for how many KS to consider and choosing KS priorities. In the case of a geographically disperse GET VPN network (for example, spanning multiple continents), deploying a KS in each geographical area to handle the registration is preferred to reduce registration latency.

4.2.1.1.1 KS Placement

An important consideration in the DC is the placement of the KS. The KS router is usually a "router-on-a-stick". The only function of these routers is to handle the GDOI functionality.

Figure 4-3 shows one of the ways in which the KS can be placed in the DC.

Figure 4-3: DC with Deployed KS



The KS are connected directly to the WAN aggregation devices in parallel to the GMs. This allows the KS to take advantage of the security policies placed on these aggregation devices.

4.2.1.1.2 KS Redundancy

As shown in Figure 4-3, each KS should have multiple links to the WAN aggregation devices. This provides for redundant links. In addition to that, multiple KSs can be placed in the DC. See 3.7.3.2 “Setting KS Priority” for recommendations for KS priority selection.

4.2.1.1.3 KS Loopback interface and Routing

For a KS, it is recommended to always use a loopback interface as the KS IP address for the GDOI protocol. The configuration below shows how to use the loopback interface as the server IP address in the KS configuration.

```
!
interface Loopback0
 ip address 1.2.3.4 255.255.255.255
!
crypto gdoi group <groupname>
 server local
  address ipv4 1.2.3.4
!
```

Some consideration must be given to the routing that is configured at the DC. The KS IP address, that is, the loopback interface, might need to be distributed into routing protocol at the DC.

Another factor to note while considering the routing at the KS is network convergence. When a KS initializes, it goes through a COOP election process. If network convergence takes longer than the COOP process, the KS elects itself as a primary KS, emulating a false network split (see 3.7.4.2.2 "Network Merge").

One recommendation for reducing network convergence is to enable Port Fast spanning-tree mode on the switch port that the KS connects to. A lengthy convergence time is commonly attributed to introducing a new route or prefix into the dynamic routing protocols. If the prefix associated with the KS loopback interface is persistently advertised into the network, the delayed reachability interval between the KS is dramatically shortened.

4.2.1.1.4 KS Firewall Considerations

If the DC WAN aggregation routers have a perimeter firewall used to protect the DC from the WAN, a separate DMZ can be created on the firewall for the KS. This enables the KS to remain isolated from other devices and improves security.

The following sample configuration on the FWSM creates a separate interface for the KSs.

```
!
!Vlan13 terminates KS connections
!
interface Vlan13
 nameif key_servers
 security-level 75           <-- security level lower than inside network

 ip address 133.151.1.1 255.255.255.0
!
```

The firewall access control lists (ACLs) must be carefully configured to permit GDOI control traffic in and out of the KSs. In addition, the ACL should allow routing traffic and other management traffic (telnet, SSH, and so on) to the KS.

The following sample ACL configuration can be used for the firewall. This is not a complete configuration, and ACL configuration varies depending on user requirements.

```
!
!Incoming ACL = KS_IN : Open ACL to allow all traffic from KS
!
!Outgoing ACL = KS_OUT : Control traffic that goes to the KS
!
access-list KS_IN extended permit ip any any
access-list KS_OUT extended permit udp any any eq 848
access-list KS_OUT extended permit ospf any any
!
access-group KS_IN in interface key_servers
access-group KS_OUT out interface key_servers
!
```

If the KSs are configured for multicast rekey, multicast-routing would need to be enabled on the firewall.

The following sample configuration on a Catalyst 6500 Firewall Services Module (FWSM) supports multicast rekey traffic.

```
!
multicast-routing
!
! RP = 10.200.200.200
pim rp-address 10.200.200.200
!
```

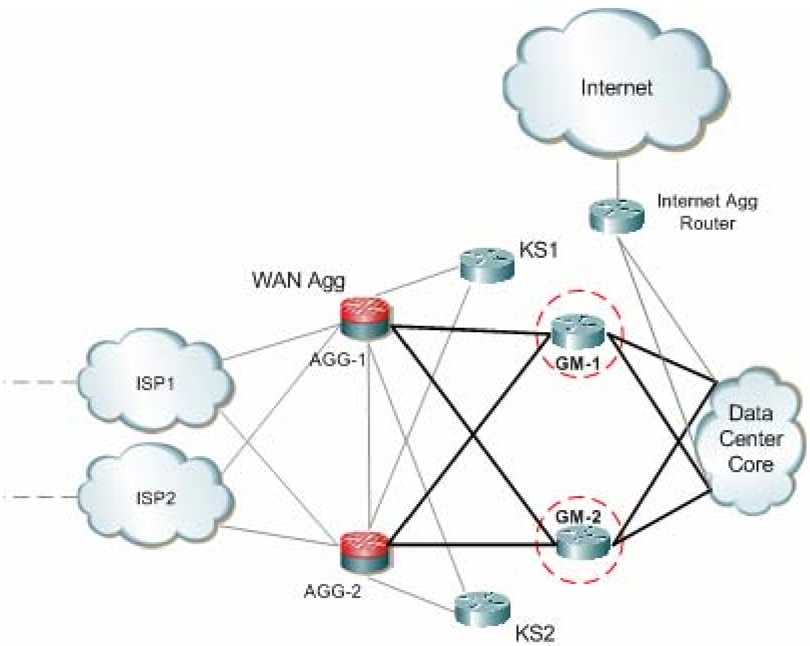

4.2.1.2 GM Design

The following section describes factors involved in GM design.

4.2.1.2.1 GM Placement

The GM handles all the encrypted traffic to and from the DC. For this reason, the GM must be placed inline with the data path just after the WAN aggregation routers. Figure 4-4 shows GM router placement.

Figure 4-4: DC with Deployed GMs



4.2.1.2.2 GM Redundancy and Load Balancing

As with other devices, it is recommended to provide multiple link redundancy to the GM. As shown in the Figure 4-4, each GM has dual links to the WAN aggregation devices and to the DC core. As an example, in the DC shown in Figure 4-4, GM1 has two routes to either KS.

```
GM1#sh ip route 139.1.1.1          <-- KS1 = 139.1.1.1
Routing entry for 139.1.1.1/32
  Known via "ospf 10", distance 110, metric 14, type intra area
  Last update from 133.150.0.254 on GigabitEthernet0/1, 1w1d ago
  Routing Descriptor Blocks:
  * 136.150.0.254, from 139.1.1.1, 1w1d ago, via GigabitEthernet0/3
      Route metric is 14, traffic share count is 1
                                         <-- G0/3 is link to AGG-2
    133.150.0.254, from 139.1.1.1, 1w1d ago, via GigabitEthernet0/1
      Route metric is 14, traffic share count is 1
                                         <-- G0/1 is link to AGG-1
```

A DC typically carries high throughput traffic. To handle the DC traffic load, it is recommended to deploy multiple GMs. This provides for redundancy and load-balancing. Routing must be designed carefully at the DC so that GMs have the same routing knowledge to provide for load balancing. Figure 4-4 shows how two GMs are deployed.

The following CLI snippets demonstrate the load balancing capability of the DC.

Consider an example in which network 151.1.6.0/24 is a destination available on the private network, reachable either through ISP1 or ISP2.

As seen from a DC router R1, multiple routes are available for the destination network.

```
R1#sh ip route 151.1.6.0
Routing entry for 151.1.6.0/24
  Known via "ospf 10", distance 110, metric 10
  Tag 500, type extern 2, forward metric 2
  Last update from 10.1.1.1 on GigabitEthernet0/1, 6d21h ago
  Routing Descriptor Blocks:
    10.2.1.1, from 133.150.2.1, 6d21h ago, via GigabitEthernet0/2
      Route metric is 10, traffic share count is 1
      Route tag 500
                                                 <-- G0/2 is link to GM-2
  * 10.1.1.1, from 134.202.1.254, 6d21h ago, via GigabitEthernet0/1
      Route metric is 10, traffic share count is 1
      Route tag 500
                                                 <-- G0/1 is link to GM-1
```

The Cisco Express Forwarding (CEF) table for the destination can also verify load balancing. As seen in the show command below, the load distribution is 01 01 01 … and **`traffic share 1`** shows equal cost per destination load-sharing at the access-layer router.

```
R1#sh ip cef 151.1.6.0 internal
151.1.6.0/24, version 21109, epoch 0, per-destination sharing
0 packets, 0 bytes
  Flow: Origin AS 0, Peer AS 0, mask 24
  via 10.2.1.1, GigabitEthernet0/2, 0 dependencies
    traffic share 1
    next hop 10.2.1.1, GigabitEthernet0/2
    valid adjacency
  via 10.1.1.1, GigabitEthernet0/1, 0 dependencies
    traffic share 1
    next hop 10.1.1.1, GigabitEthernet0/1
    valid adjacency

  0 packets, 0 bytes switched through the prefix
  tmstats: external 0 packets, 0 bytes
           internal 0 packets, 0 bytes
  Load distribution: 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 (refcount 1)

  Hash  OK  Interface                 Address         Packets
  1     Y   GigabitEthernet0/2        10.2.1.1              0
  2     Y   GigabitEthernet0/1        10.1.1.1              0
  3     Y   GigabitEthernet0/2        10.2.1.1              0
  4     Y   GigabitEthernet0/1        10.1.1.1              0
  5     Y   GigabitEthernet0/2        10.2.1.1              0
  6     Y   GigabitEthernet0/1        10.1.1.1              0
  7     Y   GigabitEthernet0/2        10.2.1.1              0
  8     Y   GigabitEthernet0/1        10.1.1.1              0
  9     Y   GigabitEthernet0/2        10.2.1.1              0
  10    Y   GigabitEthernet0/1        10.1.1.1              0
  11    Y   GigabitEthernet0/2        10.2.1.1              0
  12    Y   GigabitEthernet0/1        10.1.1.1              0
  13    Y   GigabitEthernet0/2        10.2.1.1              0
  14    Y   GigabitEthernet0/1        10.1.1.1              0
```



```
  15    Y   GigabitEthernet0/2       10.2.1.1           0
  16    Y   GigabitEthernet0/1       10.1.1.1           0
  refcount 6
R1#
```

Tip: To see the exact route that a packet with certain source and destination IP address might take, use the command **`show ip cef exact-route <source-IP> <dest-IP>`**.

4.2.1.2.3 GM Loopback Interface and Routing

If a GM has multiple interfaces that are part of the same GDOI group, it is recommended to use a loopback interface to source the crypto. If a loopback interface is not used, each interface that handles encrypted traffic must register individually with the KS.

The KS would see these as separate requests and must keep multiple records for the same GM, which also means sending multiple rekeys. If crypto is sourced from the loopback interface instead, the GM registers only once with the KS.

The following configuration shows how this can be achieved:

```
!
interface GigabitEthernet0/1
 description *** To AGG-1 ***
 crypto map dgvpn
!
interface GigabitEthernet0/2
 description *** To AGG-2 ***
 crypto map dgvpn
!
interface Loopback0
 ip address 1.2.3.4 255.255.255.255
!
crypto map dgvpn local-address Loopback0
!
```

If using a loopback interface for crypto, routing should be designed so that GM IP address has connectivity to all KSs. This means that the GM loopback IP address must be advertised via a routing protocol.

4.2.1.2.4 Firewall Considerations for GMs

Special considerations must be taken to place the GM behind the WAN aggregation devices that implement a firewall.

Certain firewalls, such as Cisco ASA or FWSM modules (for Catalyst 6500) might not be able to handle encrypted multicast traffic, i.e., the firewall cannot forward Encapsulating Security Payload (ESP) encapsulated multicast traffic. In such scenarios, it is recommended to place the GMs outside of the firewall. Because only encrypted traffic traverses GMs, GMs need not be placed behind the firewall. This is true even without the presence of the multicast traffic. In the absence of a firewall, communication through GMs can be secured through ACLs that permit only management and encrypted traffic to and from the GM.

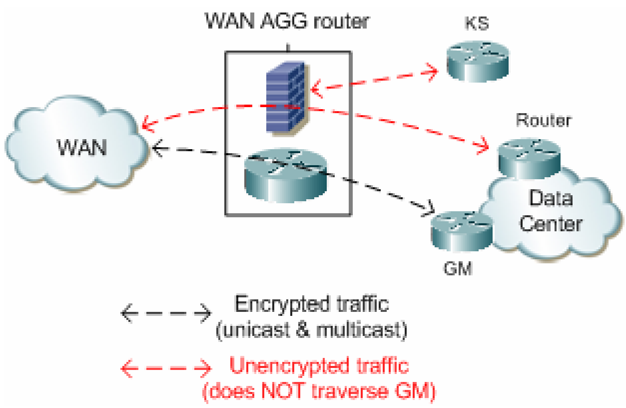
Note: ASA and FWSM do not forward multicast traffic for non-UDP encapsulation. Refer to CSCsk37000.

Figure 4-5 shows that all encrypted traffic should bypass the firewall module of the Cisco 6500 that serves as the DC WAN aggregation router, but all unencrypted traffic from the branches passes through the firewall.

Because all control plane traffic from the KSs use UDP encapsulation, the KSs can be secured behind the firewall even when multicast rekey is deployed.

Note: It is recommended to design the data center networks so that GM routers carry only encrypted traffic. This means that unencrypted traffic to and from the DC should be routed via a separate router, as shown in Figure 4-5.

Figure 4-5: Encrypted Traffic to the GM Bypasses a Firewall



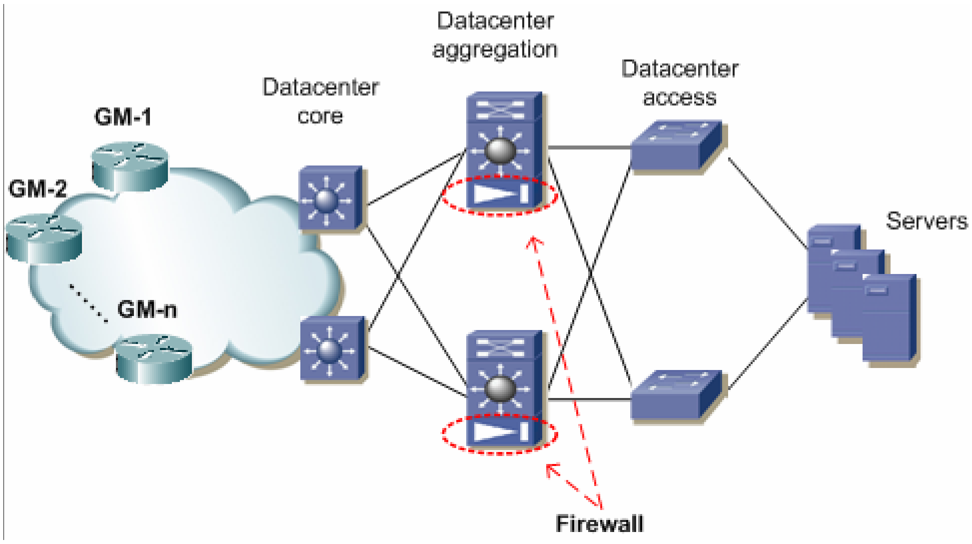
As mentioned, ACLs on the DC WAN aggregation router can protect the GMs. The following sample ACL permits ESP (encrypted traffic), UDP 848 (GDOI), OSPF (DC routing protocol), and PIM. An actual ACL will differ, depending on user requirements.

```
!
access-list 100 permit pim any any
access-list 100 permit esp any any
access-list 100 permit udp any any eq 848
access-list 100 permit ospf any any
access-list 100 deny   ip any any
!
! SVI - Vlan20 terminates GM connections on 6500
!
interface Vlan20
 ip access-group 100 out
!
```

Beyond the GM, unencrypted traffic now flows through the DC core. The firewall at the WAN aggregation cannot sniff into the encrypted traffic, so sufficient protection must be added to the DC LAN. It is recommended to deploy another firewall, such as an ASA, or provide firewall services as part of the DC aggregation layer.

Figure 4-6 shows a typical DC with firewall services as part of the DC Aggregation layer.

Figure 4-6: Additional Firewall Services Deployed in the DC



4.2.1.3 QoS

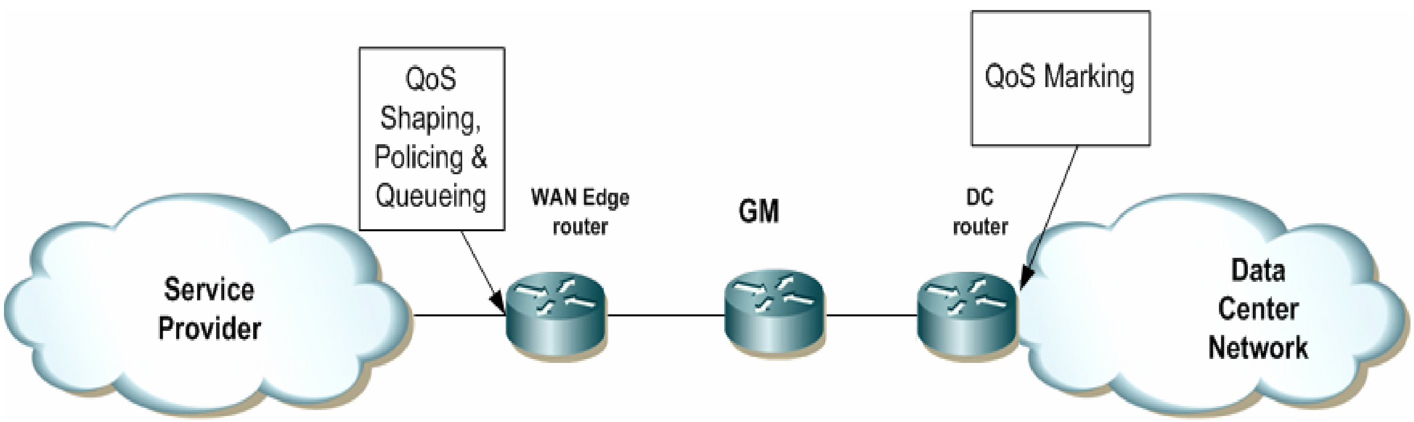
In a DC design, it is recommended not to deploy QoS functionality on the GMs to maintain optimal GM performance. If applications do not mark packets correctly, it is recommended to mark the traffic before the traffic reaches a GM (that is, at the access layer routers). Because Differentiated Services Code Point (DSCP) bits are preserved after encryption, QoS features (shaping, policing, and queuing) can easily be deployed on the aggregation routers, which typically handle QoS features in hardware. This helps free GM resources to act as a dedicated encryption device.

It is recommended to consult *Cisco Enterprise QoS Solution Reference Network Design Guide* for detailed QoS information.

http://www.cisco.com/application/pdf/en/us/guest/netsol/ns432/c649/ccmigration_09186a008049b062.pdf

For a detailed description of QoS handling with GET VPN on the same device, see 4.3.4.1 "QoS."

Figure 4-7 Recommended DC QoS Features Placement



The following sample access router configuration for marking traffic can be used as a reference.


```
! Class Map to Classify the Traffic from Data Center Network
class-map match-any INBOUND-CALL-SIGNALING
 match protocol sip
class-map match-any INBOUND-TRANSACTIONAL-DATA
 match protocol telnet
class-map match-any INBOUND-NETWORK-MANAGEMENT
 match protocol snmp
class-map match-any INBOUND-SCAVENGER
 match protocol http
class-map match-any INBOUND-VOICE
 match protocol rtp audio
class-map match-any INBOUND-MISSION-CRITICAL-DATA
 match protocol smtp
class-map match-any INTERACTIVE-VIDEO
 match ip dscp af41
class-map match-any INBOUND-INTERACTIVE-VIDEO
 match protocol rtp video
class-map match-any INBOUND-STREAMING-VIDEO
 match protocol rtsp

! Policy to Mark the Traffic From Data Center Network
policy-map LAN-INBOUND-MARKING
 class INBOUND-VOICE
  set ip dscp ef
 class INBOUND-CALL-SIGNALING
  set ip dscp ef
 class INBOUND-INTERACTIVE-VIDEO
  set ip dscp af41
 class INBOUND-STREAMING-VIDEO
  set ip dscp cs4
 class INBOUND-MISSION-CRITICAL-DATA
  set ip dscp 25
 class INBOUND-NETWORK-MANAGEMENT
  set ip dscp 25
 class INBOUND-TRANSACTIONAL-DATA
  set ip dscp af21
 class INBOUND-SCAVENGER
  set ip dscp cs1

interface FastEthernet2/0.302
 description *** Facing Data Center Network ****
 encapsulation dot1Q 302
service-policy input LAN-INBOUND-MARKING
```

On the aggregation network switch, a sample configuration is as follows. Traffic classes (TCs) are consolidated because of the limited number of classes in the switch.

```
! Class Map to Classify the Traffic Already Marked by Access Routers
class-map match-any STREAMING-VIDEO
  match ip dscp cs4
class-map match-any INTERACTIVE-VIDEO
  match ip dscp af41
class-map match-any VOICE
  match ip dscp ef
  match ip dscp af31
  match ip dscp cs3
class-map match-any MISSION-CRITICAL-DATA
  match ip dscp 25
  match ip dscp cs2
class-map match-any ROUTING
```



```
  match ip dscp cs6
class-map match-any SCAVENGER
  match ip dscp cs1
class-map match-any TRANSACTIONAL-DATA
  match ip dscp af21

! Policy Map to Do QoS Shaping,Queuing & Policing
policy-map GETVPN-QOS-POLICY
  class ROUTING
    bandwidth percent 3
  class VOICE
    bandwidth percent 15
  class INTERACTIVE-VIDEO
    bandwidth percent 15
  class STREAMING-VIDEO
    bandwidth percent 20
  class MISSION-CRITICAL-DATA
    bandwidth percent 10
    random-detect
  class TRANSACTIONAL-DATA
    bandwidth percent 10
    random-detect
  class SCAVENGER
    bandwidth percent 2
    random-detect

interface GigabitEthernet2/2/0
 description *** TO Service Provider Network***
 service-policy output GETVPN-QOS-POLICY
```

Note: 7200 NPE-G2 with VSA always add encryption overhead to the QoS calculations. Refer to CSCso69566 when deploying QoS on this platform.

4.3 Branch Design

This section provides an overview of GET VPN implementation on various branch profiles mentioned in 4.1.2 “Branch Design.”

4.3.1 Single-Tier Branch HA

In a single-tier branch design, there is one link to the MPLS provider, and GET VPN provides encryption on the MPLS link. A less expensive cable or DSL link provides HA. To securely send traffic over an Internet link, IPsec encryption must be enabled over this link, too. Because GET VPN cannot be deployed over Internet, a traditional IPsec tunneling solution must be deployed over the backup link.

Any (or all) of the following three IPsec solutions can be deployed as part of the branch security design:

- Branch with DMVPN over the backup link
- Branch with EzVPN over the backup link
- Branch with VTI over the backup link

In the following design discussions, it is assumed that:

- Tier 1 branches receive a default route from the MPLS PE using Border Gateway Protocol (BGP).
- The branch receives a dynamic address and a second default route (usually with a metric of 254) from the Internet provider
- Corporate security policy does not permit split tunneling over the Internet. That is, all traffic (including Internet traffic) must be encrypted and routed to Headquarters, where it goes through corporate firewalls.

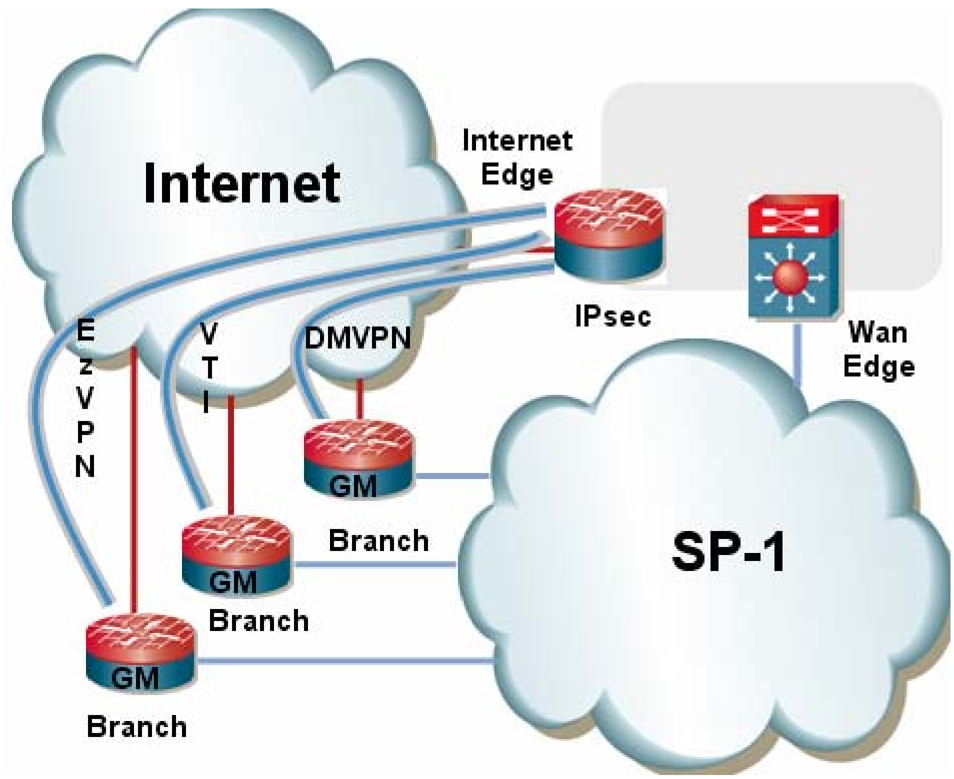
The underlying idea is straightforward: the default route over the backup link is advertised with a worse metric. If the MPLS path is unavailable, routing then directs all traffic via backup path. If the MPLS network loses connectivity (data path problem), no routing change is triggered. Cisco PfR (Performance Routing) can be deployed as discussed in 4.3.4.2 "PfR."

Note: This section does not cover branch security design in detail. Refer to SAFE documentation for design recommendations:

http://www.cisco.com/en/US/solutions/ns340/ns414/ns742/ns744/networking_solutions_products_genericcontent0900aecd805f63af.html

---

Figure 4-8 Branch Redundancy



4.3.1.1 Branch Redundancy Using DMVPN

A hub-and-spoke DMVPN network can be configured over the backup link to provide an encrypted Internet channel. The advantage of using DMVPN is its ability to set up GRE tunnels for dynamically addressed spokes, and the ability to choose the routing protocol to run over the backup interface. Routing can then be controlled according to customer requirements to provide backup connectivity. The DMVPN solution also supports multicast streaming from the hub to the spoke networks.

In the DMVPN scenario, routing is controlled as follows:

1. The Internet ISP default route (metric 254) is overwritten by the MPLS provider BGP default route (metric 20) because of preferred metric.
2. A static route for DMVPN head end is installed pointing to the gateway advertised by the Internet ISP.
3. EIGRP advertises a third default route over the DMVPN tunnel interface that has a metric better than the ISP default route (EIGRP – 90) but worse than MPLS provider default route (BGP – 20). Under normal circumstances, therefore, the default route will direct all traffic to MPLS link.
4. If the MPLS link is down and default route is withdrawn, the EIGRP default route is installed in the routing table and all traffic is directed to the tunnel, where it is encrypted using DMVPN and directed to DMVPN hub.
5. When the MPLS link comes back up, the default route learned via BGP is installed in the routing table, and traffic automatically switches back to the MPLS (primary) link.

Tip: Because DMVPN supports dynamic routing, DMVPN is the preferred backup technology.

4.3.1.1.1 Configuration on the Spoke

```
!
crypto keyring VPN
!
! Key to 181.100.100.100 is for DMVPN HUB
!
  pre-shared-key address 181.100.100.100 key ipsec123
  pre-shared-key address 0.0.0.0 0.0.0.0 key nsite123
!
! This policy is for GETVPN
!
crypto isakmp policy 10
 encr aes
 group 2
 lifetime 1200
!
! This policy is for DMVPN
!
crypto isakmp policy 20
 encr aes
 authentication pre-share
 group 5
crypto isakmp keepalive 30
!
! No need for transform-set for GETVPN. This transform-set is for DMVPN
!
crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
 mode transport
!
crypto ipsec profile dmvpn
 set transform-set AES_SHA
!
crypto gdoi group dgvpn
```



```
 identity number 61440
 server address ipv4 139.1.1.1
 server address ipv4 139.1.1.3
 server address ipv4 139.1.1.4
!
! crypto map dgvpn 10 gdoi
 set group dgvpn
!
! DMVPN Tunnel Configuration for Hub-Spoke DMVPN
!
interface Tunnel1
 bandwidth 1000
 ip mtu 1400
 ip address 172.20.1.1 255.255.255.0
 ip nhrp authentication nsite
 ip nhrp map multicast 181.100.100.100
 ip nhrp map 172.20.1.254 181.100.100.100
 ip nhrp network-id 101
 ip nhrp holdtime 1500
 ip nhrp nhs 172.20.1.254
 ip tcp adjust-mss 1360
 ip nhrp registration no-unique
 tunnel source GigabitEthernet0/0
 tunnel destination 181.100.100.100
 tunnel protection ipsec profile dmvpn
!
! Interface connecting to DSL/Cable Modem/Router
!
interface GigabitEthernet0/0
 description To Internet ISP
 ip address dhcp
 load-interval 30
 duplex full
 speed 100
!
interface GigabitEthernet0/1.203
 description LAN Interface
 encapsulation dot1Q 203
 ip address 151.1.3.254 255.255.255.0
!
interface Serial0/0/0
 description TO MPLS WAN
 bandwidth 1459
 no ip address
 encapsulation frame-relay
 service-module t1 timeslots 1-24
 frame-relay lmi-type ansi
!
interface Serial0/0/0.102 point-to-point
 ip address 134.104.1.1 255.255.255.252
 ip tcp adjust-mss 1360
 crypto map dgvpn
!
! Running EIGRP over DMVPN Tunnel
!
router eigrp 10
 network 151.1.3.0 0.0.0.255
 network 172.20.1.0 0.0.0.255
 no auto-summary
```



```
!
router bgp 106
 no synchronization
 bgp log-neighbor-changes
 network 151.1.3.0 mask 255.255.255.0
 neighbor 134.104.1.2 remote-as 500
 neighbor 134.104.1.2 update-source Serial0/0/0.102
 no auto-summary
!
! IP route for DMVPN hub pointing to DHCP
!
ip route 181.100.100.100 255.255.255.255 dhcp
!
```

4.3.1.1.2 Routing Table on the Spoke

The following show command illustrates normal behavior:

```
DMVPN-GM#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 134.104.1.2 to network 0.0.0.0

     172.20.0.0/24 is subnetted, 1 subnets
C       172.20.1.0 is directly connected, Tunnel1
     182.1.0.0/24 is subnetted, 1 subnets
C       182.1.1.0 is directly connected, GigabitEthernet0/0
     134.104.0.0/30 is subnetted, 1 subnets
C       134.104.1.0 is directly connected, Serial0/0/0.102
     181.100.0.0/32 is subnetted, 1 subnets
S       181.100.100.100 [1/0] via 182.1.1.254
     151.1.0.0/24 is subnetted, 1 subnets
C       151.1.3.0 is directly connected, GigabitEthernet0/1.203
B*   0.0.0.0/0 [20/0] via 134.104.1.2, 1w5d
```

The following show command illustrates behavior when MPLS is down:

```
DMVPN-GM#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 172.20.1.254 to network 0.0.0.0

     172.20.0.0/24 is subnetted, 1 subnets
C       172.20.1.0 is directly connected, Tunnel1
     182.1.0.0/24 is subnetted, 1 subnets
```

```
C       182.1.1.0 is directly connected, GigabitEthernet0/0
     181.100.0.0/32 is subnetted, 1 subnets
S       181.100.100.100 [1/0] via 182.1.1.254
     151.1.0.0/24 is subnetted, 1 subnets
C       151.1.3.0 is directly connected, GigabitEthernet0/1.203
D*EX 0.0.0.0/0 [170/297246976] via 172.20.1.254, 00:00:07, Tunnel1
```

4.3.1.2 Branch Redundancy Using EZVPN

Cisco IPsec EzVPN technology can be deployed over the backup link provide an encrypted Internet channel. The advantage of using EzVPN is the ability to set up IPsec tunnels for dynamically addressed spokes, and the ability scale very high. In the EzVPN solution, Reverse Route Injection (RRI) provides routing and routing protocols are not supported. EzVPN solution does not support multicast streaming. The biggest advantage of EzVPN as mentioned before is the high scalability on the EzVPN hub.

In the EzVPN scenario, routing is controlled as follows:

- The Internet ISP default route (metric 254) is overwritten by the BGP default route (metric 20) of the MPLS provider because of the preferred metric.
- A static route, pointing to the gateway advertised by the Internet ISP, is installed for the EzVPN headend.
- There is no need to advertise a third route, as is the case with DMVPN or VTI.
- If the MPLS link is down and the default route is withdrawn, the Internet ISP default route is installed in the routing table. All traffic is directed to the interface connected to Internet, where the crypto map intercepts the traffic, encrypts it, and directs it to the EzVPN hub.
- When the MPLS link recovers, traffic automatically switches back to MPLS (the primary link) because of routing.

```
!
! Key used for GETVPN
!
crypto keyring VPN
  pre-shared-key address 0.0.0.0 0.0.0.0 key nsite123
!
! Policy for GETVPN
!
crypto isakmp policy 10
 encr aes
 group 2
 lifetime 1200
!
! Policy for EzVPN
!
crypto isakmp policy 20
 encr aes
 authentication pre-share
 group 5
!
! EzVPN Client Configuration
!
crypto ipsec client ezvpn raleigh
 connect auto
 group TUC key cisco123
 mode network-extension
 peer 181.100.100.100
```



```
 username TUC3845GM password cisco
 xauth userid mode local
!
crypto gdoi group dgvpn
 identity number 61440
 server address ipv4 139.1.1.1
 server address ipv4 139.1.1.3
 server address ipv4 139.1.1.4
crypto map dgvpn 10 gdoi
 set group dgvpn
!
interface GigabitEthernet0/0
 description TO MPLS WAN
 ip address 131.102.1.1 255.255.255.0
 crypto map dgvpn
!
interface GigabitEthernet0/1.51
 description TO Internet
 encapsulation dot1Q 51
 ip address dhcp
 crypto ipsec client ezvpn Raleigh
!
interface GigabitEthernet0/1.208
 description TO LAN
 encapsulation dot1Q 208
 ip address 151.1.8.254 255.255.255.0
 crypto ipsec client ezvpn raleigh inside
!
router bgp 111
 no synchronization
 bgp log-neighbor-changes
 neighbor 131.102.1.254 remote-as 500
 neighbor 131.102.1.254 update-source GigabitEthernet0/0
 no auto-summary
!
! Route For EzVPN Hub
!
ip route 181.100.100.100 255.255.255.255 dhcp
```

4.3.1.2.1 Routing Table on the Spoke

The following show command illustrates normal behavior:

```
EZVPN-GM#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 131.102.1.254 to network 0.0.0.0

C    24.0.0.0/8 is directly connected, GigabitEthernet0/1.24
     131.102.0.0/24 is subnetted, 1 subnets
C       131.102.1.0 is directly connected, GigabitEthernet0/0
     182.1.0.0/24 is subnetted, 1 subnets
```



```
C       182.1.1.0 is directly connected, GigabitEthernet0/1.51
     181.100.0.0/32 is subnetted, 1 subnets
S       181.100.100.100 [1/0] via 182.1.1.254
     151.1.0.0/24 is subnetted, 1 subnets
C       151.1.8.0 is directly connected, GigabitEthernet0/1.208
B*   0.0.0.0/0 [20/0] via 131.102.1.254, 1w6d
```

4.3.1.2.2 Routing Table on the Spoke

The following show command illustrates behavior when MPLS is down:

```
EZVPN-GM#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 182.1.1.254 to network 0.0.0.0

C    24.0.0.0/8 is directly connected, GigabitEthernet0/1.24
     182.1.0.0/24 is subnetted, 1 subnets
C       182.1.1.0 is directly connected, GigabitEthernet0/1.51
     181.100.0.0/32 is subnetted, 1 subnets
S       181.100.100.100 [1/0] via 182.1.1.254
     151.1.0.0/24 is subnetted, 1 subnets
C       151.1.8.0 is directly connected, GigabitEthernet0/1.208
S*   0.0.0.0/0 [254/0] via 182.1.1.254

*** Proxies dictate everything must be encrypted ***


EZVPN-GM#sh crypto ipsec sa interface g0/1.51
interface: GigabitEthernet0/1.51
    Crypto map tag: GigabitEthernet0/1.51-head-0, local addr 182.1.1.25

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (151.1.8.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)

<SNIP>
```

4.3.1.3 Branch Redundancy Using VTI

Cisco IPsec VTI technology can be deployed over the backup link to provide an encrypted Internet channel. The advantage of using VTI is its ability to scale and support multicast streaming from hub to spoke.

In the VTI scenario, routing is controlled as follows:

- The Internet ISP default route (metric 254) is overwritten by the BGP default route (metric 20) of the MPLS provider because of preferred metric.
- A specific route for IPsec VTI gateway is added pointing to the Internet gateway.

- A static default route pointing to IPsec VTI is installed with a better metric than the Internet default, but worse than the BGP default.
- If the MPLS link is down and the BGP default route is withdrawn, a static default route is installed in the routing table and all the traffic is directed to VTI. All traffic is encrypted and directed to VTI hub.
- When the MPLS link recovers, the default route learned via BGP is installed in the routing table, and traffic automatically switches back to MPLS (primary) link.

```
!
! Policy for GETVPN
!
crypto isakmp policy 10
 encr aes
 group 2
 lifetime 1200
!
! Policy for VTI
!
crypto isakmp policy 20
 encr aes
 authentication pre-share
 group 5
!
! Preshared key for VTI headend
!
crypto isakmp key ipsec123 address 181.100.100.100
!
crypto isakmp key nsite123 address 0.0.0.0 0.0.0.0
crypto isakmp keepalive 30
!
!
crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
!
crypto ipsec profile VTI
 set transform-set AES_SHA
!
crypto gdoi group dgvpn
 identity number 61440
 server address ipv4 139.1.1.1
 server address ipv4 139.1.1.2
 server address ipv4 139.1.1.3
 server address ipv4 139.1.1.4
!
crypto map dgvpn 10 gdoi
 set group dgvpn
!
! Virtual Tunnel Interface
!
interface Tunnel1
 ip address 172.21.1.1 255.255.255.252
 tunnel source GigabitEthernet0/2
 tunnel destination 181.100.100.100
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile VTI
!
interface GigabitEthernet0/1
 description TO MPLS WAN
 ip address 134.108.1.1 255.255.255.0
```



```
 crypto map dgvpn
!
interface GigabitEthernet0/2
 description TO INTERNET
 ip address 182.1.1.101 255.255.255.0
!
interface FastEthernet2/0
 description TO LAN
 ip address 151.1.9.254 255.255.255.0
!
router bgp 109
 no synchronization
 bgp log-neighbor-changes
 network 151.1.9.0 mask 255.255.255.0
 neighbor 134.108.1.254 remote-as 500
 neighbor 134.108.1.254 update-source GigabitEthernet0/1
no auto-summary
!
! Default route to VTI – preferred over Internet Default Route
!
ip route 0.0.0.0 0.0.0.0 Tunnel1 100
!
ip route 181.100.100.100 255.255.255.255 182.1.1.254
```

4.3.1.3.1 Routing Table on the Spoke

The following show command illustrates normal behavior:

```
VTI-GM#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 134.108.1.254 to network 0.0.0.0

     172.21.0.0/30 is subnetted, 1 subnets
C       172.21.1.0 is directly connected, Tunnel1
     182.1.0.0/24 is subnetted, 1 subnets
C       182.1.1.0 is directly connected, GigabitEthernet0/2
     134.108.0.0/24 is subnetted, 1 subnets
C       134.108.1.0 is directly connected, GigabitEthernet0/1
     181.100.0.0/32 is subnetted, 1 subnets
S       181.100.100.100 [1/0] via 182.1.1.254
     151.1.0.0/24 is subnetted, 1 subnets
C       151.1.9.0 is directly connected, FastEthernet2/0
B*   0.0.0.0/0 [20/0] via 134.108.1.254, 1w6d
```

The following show command illustrates behavior when MPLS is down:

```
VTI-GM#show ip route

Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
```



```
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 0.0.0.0 to network 0.0.0.0

     172.21.0.0/30 is subnetted, 1 subnets
C       172.21.1.0 is directly connected, Tunnel1
     182.1.0.0/24 is subnetted, 1 subnets
C       182.1.1.0 is directly connected, GigabitEthernet0/2
     181.100.0.0/32 is subnetted, 1 subnets
S       181.100.100.100 [1/0] via 182.1.1.254
     151.1.0.0/24 is subnetted, 1 subnets
C       151.1.9.0 is directly connected, FastEthernet2/0
S*   0.0.0.0/0 is directly connected, Tunnel1
VTI-GM#
```

4.3.2 Dual-Tier Branch HA

As described in 4.1.2 "Branch ," there are three typical dual-tier branch profiles. This section provides an overview of GET VPN implementations on dual-tier branches. In the following section, the same GDOI policy (GMs registering to the same GDOI group) is pushed down the GMs on both provider links.

It is possible, however, that GMs register to different KSs (or groups) and download different GET VPN policies on different providers (one for Provider-A and one for Provider-B). In this case, redundant providers and security systems are deployed on the two links.

4.3.2.1 Dual WAN Termination on a Single Branch Router (GM)

In this profile, a single router (GM) terminates two WAN circuits to provide link redundancy.

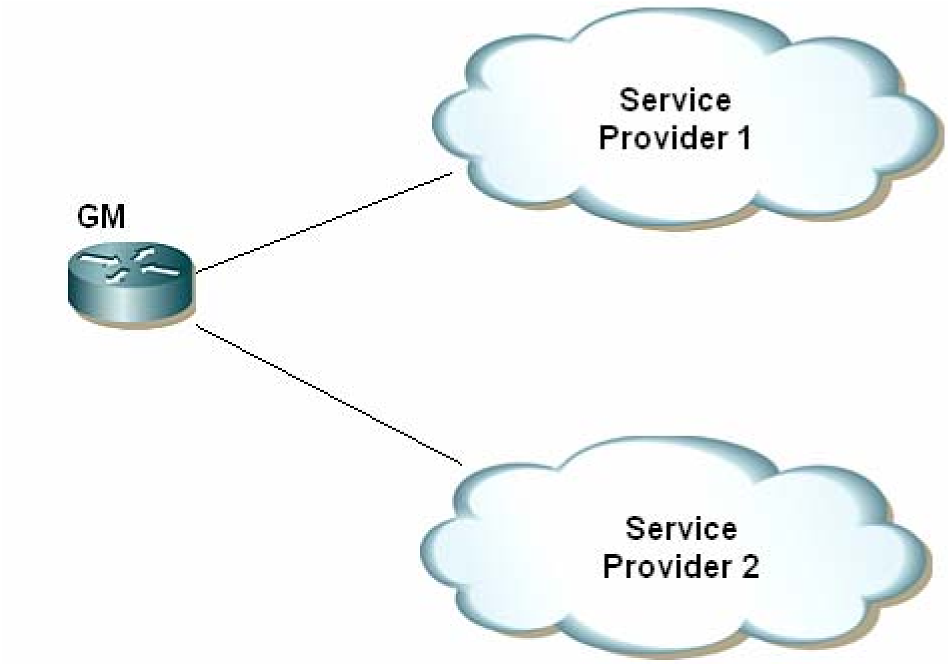
It is recommended to use the loopback address as a registration address when attaching the same crypto map on the two redundant links. Otherwise, a KS sees this single GM as two distinct GMs.

```
crypto map crypto_map_name local-address Loopback0
```

A sample configuration for this kind of branch is as following:

```
crypto gdoi group dgvpn
 identity number 61440
 server address ipv4 139.1.1.1
!
crypto map dgvpn local-address Loopback0

crypto map dgvpn 10 gdoi
 set group dgvpn

interface GigabitEthernet0/0
 description ***To Service Provider 1****
 <..>
 crypto map dgvpn
!
interface GigabitEthernet0/1
 description **** TO Service Provider 2 ****
<..>
 crypto map dgvpn
```

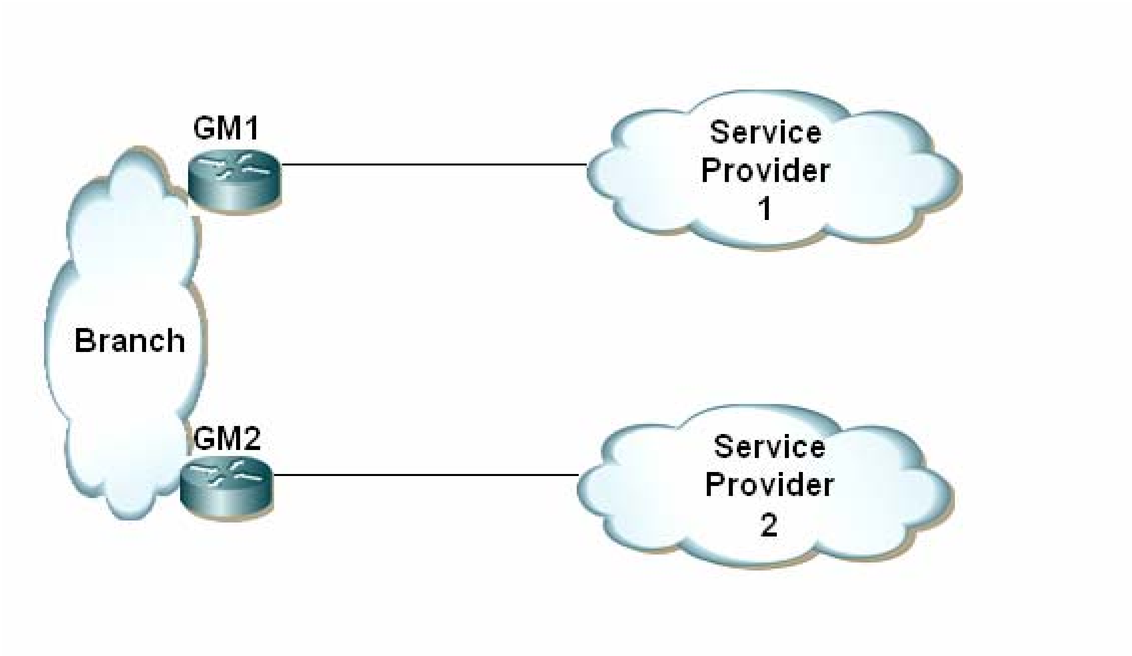

Figure 4-9 Dual WAN Termination on a Single Branch Router



4.3.2.2 Dual WAN Termination on Dual Branch Routers (GMs)

The most likely scenario to achieve dual tier branch redundancy is to use two branch routers (GMs), each connected to a different service provider (SP), so that losing a single link or GM does not impact site connectivity. Unlike traditional IPsec solutions, GET VPN GMs share the same SAs and policies. Therefore, the branch can handle asymmetric routing.

Figure 4-10 Dual Tier Branch – CEs Acting as GM



A sample follows:

GM1

```
crypto gdoi group dgvpn
 identity number 61440
 server address ipv4 139.1.1.1

crypto map dgvpn 10 gdoi
 set group dgvpn

interface GigabitEthernet0/1
 description *** To Service Provider 1 ***
crypto map dgvpn
```

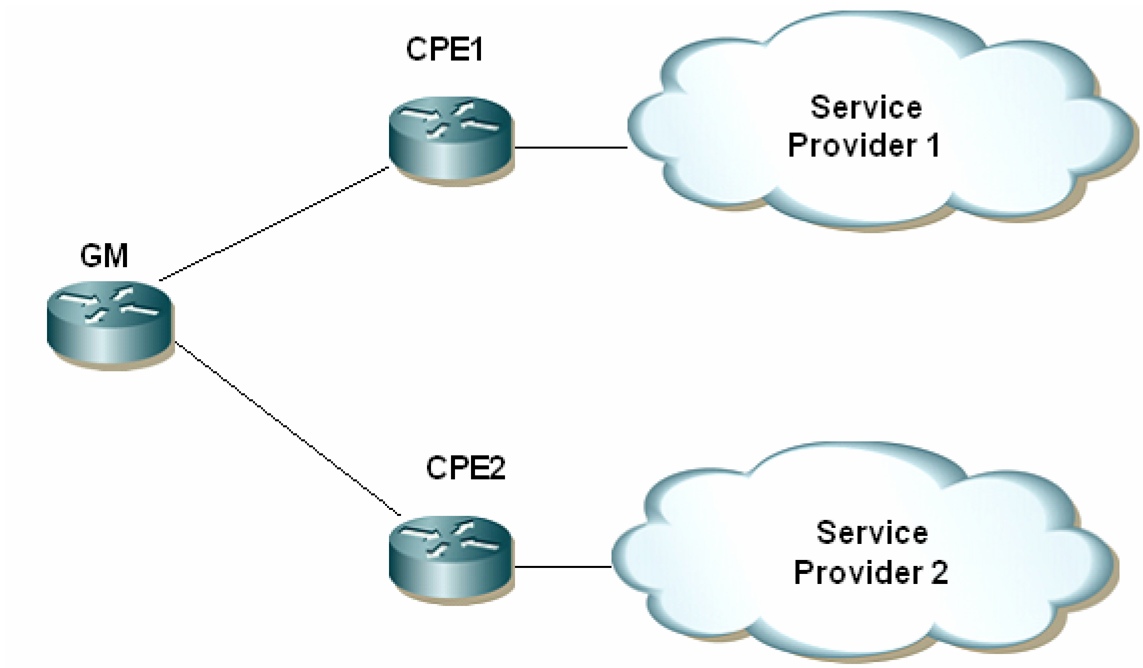
GM2

```
crypto gdoi group dgvpn
 identity number 61440
 server address ipv4 139.1.1.1

crypto map dgvpn 10 gdoi
 set group dgvpn

interface GigabitEthernet0/1
 description *** To Service Provider 2 ***
crypto map dgvpn
```

4.3.2.3 Dual WAN Termination on Provider CPE

In this case, the customer is provided two CPEs for connecting to two different service provider networks, and the customer cannot configure GET VPN on the CPEs. An enterprise-owned branch router acts as a GET VPN GM and connects to both CPEs.

Figure 4-11 Dual Tier Branch – GM behind SP CPEs



A sample configuration follows. The configuration has the same design consideration as 4.3.2.1 "Dual WAN Termination on a Single Branch Router (GM)."

```
crypto gdoi group dgvpn
 identity number 61440
 server address ipv4 139.1.1.1
!
crypto map dgvpn local-address Loopback0

crypto map dgvpn 10 gdoi
 set group dgvpn

interface GigabitEthernet0/0
 description ***To Service Provider 1 CPE****
 <..>
 crypto map dgvpn
!
interface GigabitEthernet0/1
 description **** TO Service Provider 2 CPE****
<..>
 crypto map dgvpn
```

4.3.3 Multitier Branch Redundancy

Multitier Branch HA is no different than that of the dual-tier branch from GET VPN deployment point of view. The design consideration is the same as that of dual tier branch described in 4.3.2.2 "Dual WAN Termination on Dual Branch Routers (GMs)."

4.3.4 Common GET VPN Branch Deployment Features

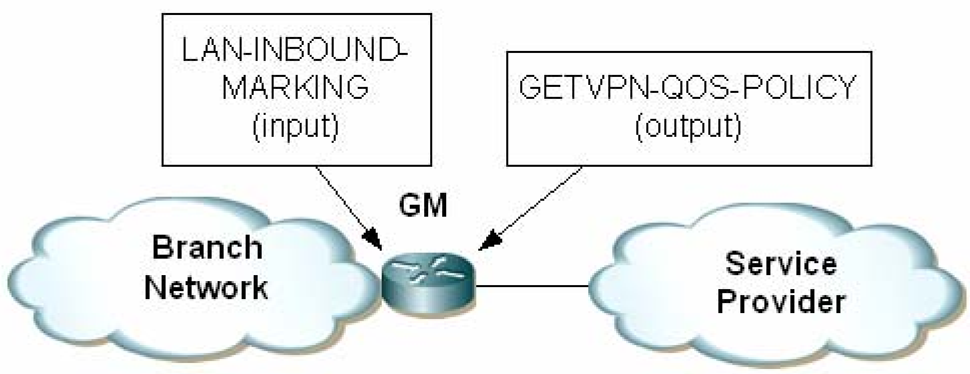
Common GET VPN branch deployment features include:

- QoS
- Performance routing (PfR)
- Wide Area Application Services (WAAS)

4.3.4.1 QoS

In GET VPN branch designs, we recommend that IP marking be done either by application or branch LAN facing interface while shaping and queuing features to be deployed together on GET VPN crypto map attached interface. Figure 4-12 shows that inbound traffic from branch LAN network is marked by the input QoS policy, while outbound traffic is handled by the output QoS policy.

Figure 4-12 Sample Branch QoS Design



4.3.4.1.1 QoS Preclassify

In every IOS IPsec solution, DSCP or TOS markings are preserved during encryption process, that is, DSCP values are copied to the new outside header. If the packets were already marked with appropriate DSCP bits, QoS can be applied on the WAN interface based on these markings. IOS provides the flexibility to apply QoS policies on the WAN interface based on original IP header or Layer 4 information (TCP/UDP ports, and so on). This can be achieved by using a **`qos pre-classify`** command in the crypto map.

A network designer must be careful when choosing to use the **`qos pre-classify`** command. The packet processing order differs when this command in the configuration:

- With **`qos pre-classify`**: 1. Classification 2. Marking 3. Policing 4. Encryption

- Without **`qos pre-classify`**: 1. Encryption 2. Classification 3. Marking 4. Policing

Also, this change of order changes how the “show policy-map interface” displays the offered rate for a particular class. The following example shows the difference in display output with exactly the same traffic.

- With **`qos pre-classify`**, the offered rate is clear traffic rate without any encryption overhead.

```
Class-map: VOICE (match-any)
     18361 packets, 3745644 bytes
     30 second offered rate 244000 bps, drop rate 45000 bps
    Match: ip dscp ef (46)
        18361 packets, 3745644 bytes
       30 second rate 244000 bps
      Queueing
       Strict Priority
       Output Queue: Conversation 72
        Bandwidth 18 (%)
        Bandwidth 276 (kbps) Burst 6900 (Bytes)
       (pkts matched/bytes matched) 18361/4920748
       (total drops/bytes drops) 2560/686080
```

- Without **`qos pre-classify`**, the offered rate is the encrypted rate with IPSec encryption overhead included.

```
 Class-map: VOICE (match-any)
      16651 packets, 4462468 bytes
      30 second offered rate 315000 bps, drop rate 45000 bps
      Match: ip dscp ef (46)
        16651 packets, 4462468 bytes
        30 second rate 315000 bps
      Queueing
        Strict Priority
        Output Queue: Conversation 72
        Bandwidth 18 (%)
        Bandwidth 276 (kbps) Burst 6900 (Bytes)
        (pkts matched/bytes matched) 16651/4462468
        (total drops/bytes drops) 2317/620956
```

Tip: If the QoS requirements can be satisfied by applying QoS policies based on DSCP/TOS bits, configuring qos pre-classify is not required.

4.3.4.1.2 Sample QoS Configuration

A sample QoS design and configuration follows:

```
! For Classification of Inbound Traffic From LAN side
class-map match-any INBOUND-CALL-SIGNALING
 match protocol sip
class-map match-any INBOUND-TRANSACTIONAL-DATA
 match protocol telnet
class-map match-any INBOUND-NETWORK-MANAGEMENT
```



```
 match protocol snmp
class-map match-any INBOUND-SCAVENGER
 match protocol http
class-map match-any INBOUND-VOICE
 match protocol rtp audio
class-map match-any INBOUND-MISSION-CRITICAL-DATA
 match protocol smtp
class-map match-any INTERACTIVE-VIDEO
 match ip dscp af41
class-map match-any INBOUND-INTERACTIVE-VIDEO
 match protocol rtp video
class-map match-any INBOUND-STREAMING-VIDEO
 match protocol rtsp

! Policy For IP Marking On Traffic From LAN Side
policy-map LAN-INBOUND-MARKING
 class INBOUND-VOICE
  set ip dscp ef
 class INBOUND-CALL-SIGNALING
  set ip dscp ef
 class INBOUND-INTERACTIVE-VIDEO
  set ip dscp af41
 class INBOUND-STREAMING-VIDEO
  set ip dscp cs4
 class INBOUND-MISSION-CRITICAL-DATA
  set ip dscp 25
 class INBOUND-NETWORK-MANAGEMENT
  set ip dscp 25
 class INBOUND-SCAVENGER
  set ip dscp cs1
 class INBOUND-TRANSACTIONAL-DATA
  set ip dscp af21

! For Classification of Outbound Traffic Toward WAN
class-map match-any VOICE
 match ip dscp ef
 match ip dscp af31
 match ip dscp cs3
class-map match-any STREAMING-VIDEO
 match ip dscp cs4
class-map match-any MISSION-CRITICAL-DATA
 match ip dscp 25
 match ip dscp cs2
class-map match-any ROUTING
 match ip dscp cs6
class-map match-any SCAVENGER
 match ip dscp cs1
class-map match-any TRANSACTIONAL-DATA
 match ip dscp af21

! Policy for QoS Shaping and Queueing toward WAN
policy-map GETVPN-QOS-POLICY
 class ROUTING
  bandwidth percent 3
 class VOICE
  priority percent 15
 class INTERACTIVE-VIDEO
  priority percent 15
 class STREAMING-VIDEO
  bandwidth percent 20
```



```
 class MISSION-CRITICAL-DATA
  bandwidth percent 10
  random-detect
 class TRANSACTIONAL-DATA
  bandwidth percent 10
  random-detect
 class SCAVENGER
  bandwidth percent 2
  random-detect

! Policy Attached to WAN Facing Interface to Do Shaping and Queueing
interface GigabitEthernet0/1
 description *** To Service Provider ***
 <…>
 crypto map dgvpn
 service-policy output GETVPN-QOS-POLICY

! Policy Attached to LAN Facing Interface to Do IP Marking
interface FastEthernet2/0
 description *** TO Branch LAN network****
<…>
 service-policy input LAN-INBOUND-MARKING
```

4.3.4.2 PfR

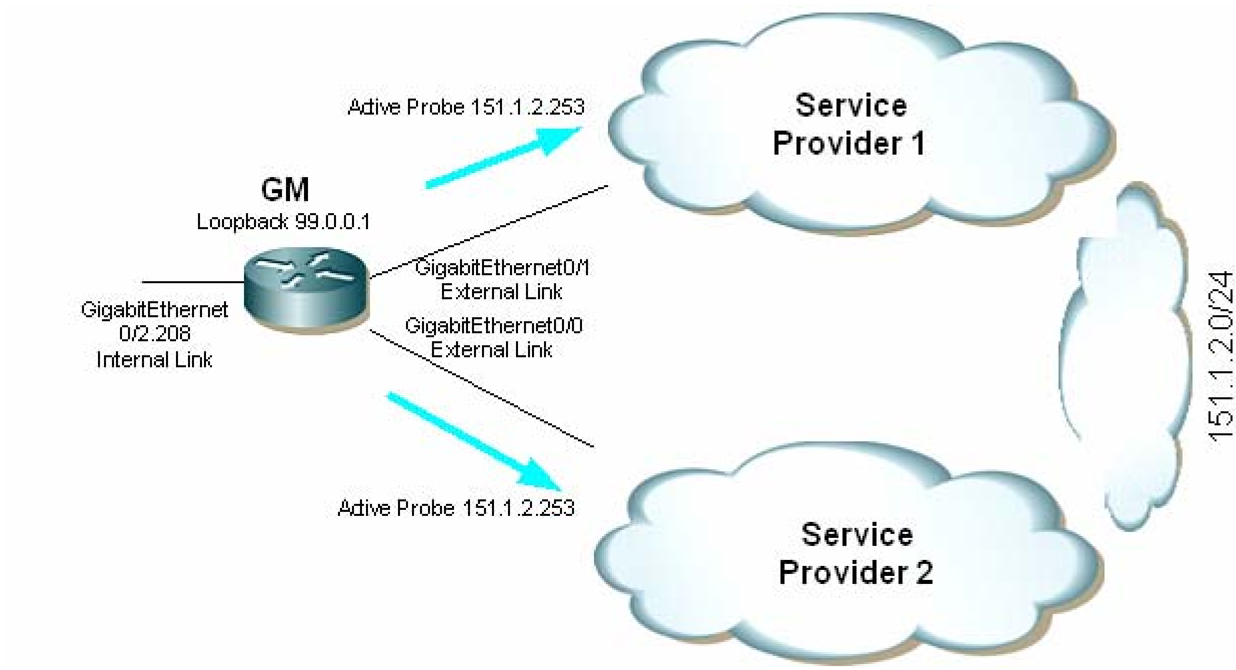
Because GET VPN dual-tier branches have multiple paths, maintaining end-to-end connectivity is vital in enterprise network operation. Cisco PfR (Performance Routing) technology can be deployed in branches with redundant links to route the traffic based on various network parameters such as delay, jitter, throughput etc and to overcome path failures. In this design we mainly use PfR to detect path failure and to maintain end-to-end connectivity.

The prerequisite of this deployment is that a branch site should have at least two redundant links on two different subnets as PfR external links, and one subnet for internal link. For more information regarding PfR deployment, refer to http://www.cisco.com/go/pfr.

In this example, as shown in Fig 4-14, PfR is deployed on a single branch router having dual WAN termination. PfR master router and PfR border router are on the same router, which also provides GET VPN functionality. PfR fast-monitoring is used for monitoring connectivity to a critical site that hosts the LAN 151.1.2.0/24. To do this, an active probe is configured for one host address (151.1.2.253) across the MPLS network.

PfR passive monitoring is used for all other traffic on the external interfaces. Path failure for active monitored traffic is detected by loss of active probes, while throughput measurements determine path failure for passive monitored traffic. If the network loses the in-use path, traffic is rerouted to the alternate path.

Figure 4-13 PfR in the GET VPN Environment




```
!KEY for auth between master and border OER router
key chain BR1
 key 1
  key-string KEYSTRING

!-----OER Master Router Configruation-----
oer master
 !oer-map DATACENTER defines special treatment for specific traffic
 policy-rules DATACENTER
 !
 logging
 !
 border 99.0.0.1 key-chain BR1
  interface GigabitEthernet0/2.208 internal
  interface GigabitEthernet0/1 external
  interface GigabitEthernet0/0 external
 !
 learn
  throughput
  periodic-interval 0
  monitor-period 1
  prefixes 250
  aggregation-type prefix-length 16
 no max range receive
 unreachable threshold 10
 mode route control
 mode monitor passive
!
!----OER Border Router Configuration----
oer border
 local Loopback99
 master 99.0.0.1 key-chain BR1
!
!Oer-map defines fast monitoring for traffic matching a prefix-list
```



```
oer-map DATACENTER 10
 match traffic-class prefix-list OER
 set mode select-exit good
 set mode route control
 set mode monitor fast
 set active-probe echo 151.1.2.253
 set probe frequency 10
!
!prefix-list used by oer-map
!network 151.1.2.0/24 is Datacenter LAN network, and hence critical
ip prefix-list OER seq 5 permit 151.1.2.0/24
!
```

To verify OER master operation, the **show oer master** command can be used. Once operational, OER master status should show as ACTIVE.

```
GM#sh oer master
OER state: ENABLED and ACTIVE
  Conn Status: SUCCESS, PORT: 3949
  Version: 2.1
  Number of Border routers: 1
  Number of Exits: 2
  Number of monitored prefixes: 2 (max 5000)
  Max prefixes: total 5000 learn 2500
  Prefix count: total 2, learn 1, cfg 1

Border           Status   UP/DOWN             AuthFail  Version
99.0.0.1         ACTIVE   UP       1d20h          0  2.1

<..>

Learn Settings:
  current state : STARTED
  time remaining in current state : 117 seconds
  throughput
  no delay
  no inside bgp
  no protocol
  monitor-period 1
  periodic-interval 0
  aggregation-type prefix-length 16
  prefixes 250
  expire after time 720
GM#
```

To verify the OER prefixes created by the OER master, use the command **show oer master prefix**. Fast monitoring creates its own prefix based on the prefix-list configured for the oer-map. In our example, this would be 151.1.2.0/24.

Passive monitoring creates its prefixes based on the traffic flow. In our example, the GM was carrying traffic to a destination 151.1.6.0/24, but the aggregation-type is configured for a prefix-length of 16. Therefore, passive monitoring creates the prefix 151.1.0.0/16.

```
GM#show oer master prefix
OER Prefix Statistics:
 Pas - Passive, Act - Active, S - Short term, L - Long term, Dly - Delay
(ms),
 P - Percentage below threshold, Jit - Jitter (ms),
 MOS - Mean Opinion Score
```



```
 Los - Packet Loss (packets-per-million), Un - Unreachable (flows-per-
million),
 E - Egress, I - Ingress, Bw - Bandwidth (kbps), N - Not applicable
 U - unknown, * - uncontrolled, + - control more specific, @ - active
probe all
 # - Prefix monitor mode is Special, & - Blackholed Prefix
 % - Force Next-Hop, ^ - Prefix is denied

Prefix              State     Time Curr BR         CurrI/F         Protocol
                 PasSDly  PasLDly   PasSUn   PasLUn  PasSLos  PasLLos
                 ActSDly  ActLDly   ActSUn   ActLUn      EBw      IBw
                 ActSJit  ActPMOS  ActSLos  ActLLos
--------------------------------------------------------------------------
151.1.0.0/16        INPOLICY        0 99.0.0.1        Gi0/0           BGP
                          U        U        0        0        0        0
                          N        N        N        N        1        1
                          N        N
151.1.2.0/24        INPOLICY       @0 99.0.0.1        Gi0/0           BGP
                          U        U        0        0        0        0
                          1        1        0        0        0        0
```

Use the following show command to display the active probes on OER:

```
GM#show oer border active-probes
        OER Border active-probes
Type      = Probe Type
Target    = Target IP Address
TPort     = Target Port
Source    = Send From Source IP Address
Interface = Exit interface
Att       = Number of Attempts
Comps    = Number of completions
N - Not applicable

Type  Target       TPort Source        Interface     Att    Comps   DSCP
echo  151.1.2.253      N 133.0.1.1     Gi0/1         173      171   0
echo  151.1.2.253      N 132.0.1.1     Gi0/0         173      173   0
```

In the preceding example, OER is configured to control routing using the command **mode route control** for both passive and active monitoring. BGP was used in the example.

Use the following show command to display OER route information:

```
GM#show oer border route bgp
BGP table version is 2366, local router ID is 132.50.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
OER Flags: C - Controlled, X - Excluded, E - Exact, N - Non-exact, I -
Injected

   Network          Next Hop        OER    LocPrf Weight Path
*> 151.1.2.0/24     133.0.1.254     CEI                0 500 ?

!
!---After one path loses connectivity, it seen that the next hop changes--
!
GM#show oer border routes bgp
```

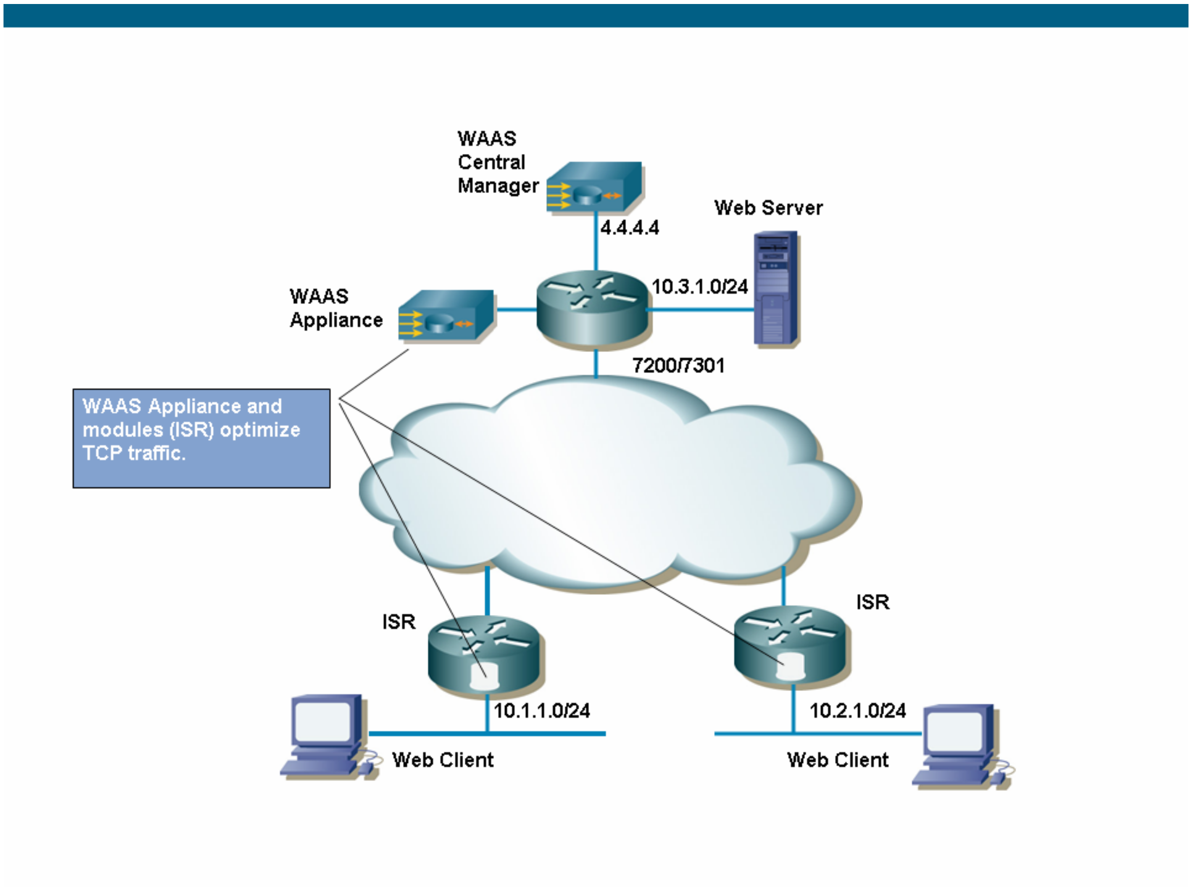
```
BGP table version is 4981, local router ID is 132.50.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
OER Flags: C - Controlled, X - Excluded, E - Exact, N - Non-exact, I -
Injected

   Network          Next Hop        OER    LocPrf Weight Path
*> 151.1.2.0/24     132.0.1.254     CEI               0 500 ?
```

4.3.4.3 WAAS

The Cisco WAAS portfolio of technologies and products give branch and remote offices LAN-like access to centrally hosted applications, servers, and storage. The solution provides powerful application delivery, acceleration, and WAN optimization to optimize the performance of any TCP-based application in a WAN environment. The Cisco WAAS solution can be deployed in a GET VPN environment by using a WAAS appliance or network modules (ISR routers).

To configure WAAS services on a GET VPN GM, we must configure the GM and the WAAS module (or appliance) attached to the GM. The WAAS solution also requires an appliance to be configured as a Central Manager (CM). A CM is typically deployed in the DC and monitors and manages the appliances and modules. To optimize TCP based traffic between a branch and the DC, the WAAS solution can be deployed as shown in Figure 4-14:

Figure 4-14 WAAS Deployment in a GET VPN Environment



4.3.4.3.1 Router Configuration

The following configuration is required on the router in addition to the GET VPN and routing configuration, All WAAS modules and appliances should be able to reach each other and the Central Manager, encrypted or in the clear.

```
! Enable TCP promiscuous service groups
! 61-Source IP Hash; 62 – Destination IP Hash
!
ip wccp 61
ip wccp 62
!
! Apply WCCP redirection on WAN Interface
!
interface GigabitEthernet0/0
 ip address 1.1.1.1 255.255.255.0
 ip wccp 62 redirect in
 crypto map getvpn
!
! Apply WCCP redirection on the LAN interface
!
interface GigabitEthernet0/1.11
 ip address 10.1.1.254 255.255.255.0
 ip wccp 61 redirect in
!
```



```
! Service Interface automatically configured for ISR modules
! This configuration is not required when using Appliance
! Address and default gateway assigned to WAE module
!
interface Integrated-Service-Engine3/0
 ip address 192.1.1.254 255.255.255.0
 ip wccp redirect exclude in
 service-module ip address 192.1.1.1 255.255.255.0
 service-module ip default-gateway 192.1.1.254
 no keepalive
!
```

4.3.4.3.2 Module/Appliance Configuration

For the first configuration of the appliance, use the console connection to connect to the device and use the username **admin** with a password of **default**. To configure WAAS modules in ISR routers, telnet to the address assigned to the service interface (shown previously) and use **admin** and **default** as username and password.

```
! WAAS version 4.0.13 (build b23 Sep  8 2007)
!
hostname 3845-WAE
!
primary-interface GigabitEthernet 1/0
!
interface GigabitEthernet 1/0
 ip address 192.1.1.1 255.255.255.0
 no autosense
 bandwidth 1000
 full-duplex
 exit
!
! The second Ethernet interface on the WAAS appliance is typically used
! as the management interface. It is not being used in this configuration
!
interface GigabitEthernet 2/0
 shutdown
 exit
!
ip default-gateway 192.1.1.254
!
no auto-register enable
!
! ip path-mtu-discovery is disabled in WAAS by default
!
ntp server 10.3.1.254
!
wccp router-list 1 192.1.1.254
wccp tcp-promiscuous router-list-num 1
wccp version 2
!
! The following configuration must be added if planning to use GRE return
!
egress-method negotiated-return intercept-method wccp
!
central-manager address 4.4.4.4
!
```


4.3.4.3.3 WAAS Configuration Verification

The following commands can be used to verify the operation of the WAAS solution in a GET VPN environment:

```
3845-Spoke#sh ip wccp inter detail

WCCP interface configuration details:
    GigabitEthernet0/1.11
        Output services: 0
        Input services:  1
        Static:          None
        Dynamic:         061
        Mcast services:  0
        Exclude In:      FALSE

    Integrated-Service-Engine3/0
        Output services: 0
        Input services:  0
        Mcast services:  0
        Exclude In:      TRUE

    Tunnel1
        Output services: 0
        Input services:  1
        Static:          None
        Dynamic:         062
        Mcast services:  0
        Exclude In:      FALSE
3845-Spoke#telnet 192.1.1.1
Cisco Wide Area Application Services Engine
3845-WAE login: admin
Password: default

3845-WAE#sh tfo connection summ

Optimized Connection List
Policy summary order: Our's, Peer's, Negotiated, Applied
F: Full optimization, D: DRE only, L: LZ Compression, T: TCP Optimization

Local-IP:Port         Remote-IP:Port        ConId    PeerId             Policy
10.1.1.10:3863        10.2.1.10:80          77       00:14:5e:83:03:b3 F,F,F,F
```

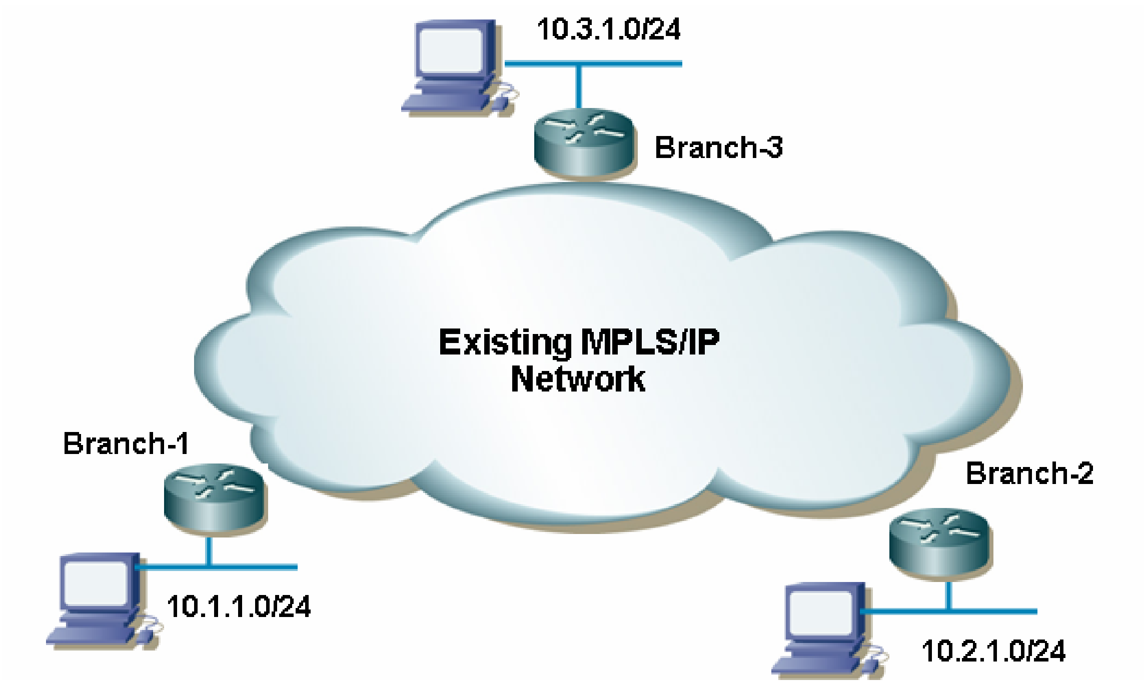
In addition to the preceding commands, **show interface** counters can compare LAN and WAN data rates. Typically, the WAN traffic rate is a fraction of LAN data rate. In addition to improving latency and response times, WAAS reduces (compresses) TCP traffic flowing through the router to improve WAN utilization and CPU overhead due to the encryption process (less traffic to encrypt).

Note: When using GRE return in WAAS solution, use a version with fix for CSCsm35350

4.4 Deploying GET VPN

A GET VPN network is seldom deployed in a new customer deployment. Typically a GET VPN deployment is over an existing IP or MPLS network. In traditional IPsec deployments, this transition is relatively simple because encryption policies explicitly define the source and destination addresses of the networks requiring encryption. Consider the following network on which encryption services must be enabled:

Figure 4-15 Existing MPLS Connected Branches



4.4.1 Traditional IPsec Deployment

In traditional IPsec, encryption policies (what must be encrypted) are defined by explicit ACL entries. This means that on Branch-1, under the crypto map, the ACL must include the source and destination prefixes of the networks which need to be encrypted and will be defined as:

```
access-list 101 permit ip 10.1.1.0 0.0.0.255 10.2.1.0 0.0.0.255
```

On Branch-2, the ACL is a mirror image:

```
access-list 101 permit ip 10.2.1.0 0.0.0.255 10.0.1.0 0.0.0.255
```

Because the source and destination entries shown above are explicit, encryption takes place only between Branch-1 and Branch-2 networks. Traffic to and from the Branch-3 network (10.3.1.0/24) is sent and received in the clear and is not impacted by the newly defined encryption policy.

This makes migration from a unencrypted network to an encryption-enabled network relatively simple. Migration can easily take place in phases, where a few branches are converted to encryption-enabled branches at a time.

4.4.2 GET VPN

In GET VPN deployments, it is recommended to keep the encryption policy (what needs to be encrypted) compact by summarizing the source and destination addresses in the encryption domain to a few entries. It means in the above example, ACL defining prefixes will be defined as:

```
access-list 101 permit ip 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
```

This ACL entry will result in only one pair of IPsec SAs in the GMs. The problem with this approach is as soon as one of the branches, e.g. Branch-1, is configured as GET VPN GM and the branch successfully registers with the KS, the branch starts encrypting all the traffic to and from the 10.0.0.0/8 network and is essentially cut off from the whole network. This makes phased migration of the network devices impossible without major disruptions in preexisting networks.

4.4.3 GET VPN Operating Modes

GET VPN allows different types of special encryption, decryption and forwarding modes for IPsec SAs for ease of deployment. It is important to understand these modes of GM operation in GET VPN to come up with a successfully migration strategy. Since the encryption and decryption behavior of the GET VPN devices is different in each mode, Table 4-1 summarizes this behavior:

Table 4-1 GET VPN Operating Modes

Conformant (Normal)	Receiving	Encrypted traffic matching policy is decrypted Encrypted traffic not matching policy is dropped Clear traffic matching policy is dropped Clear traffic not matching policy is forwarded
	Sending	Clear traffic matching policy is encrypted Clear traffic not matching policy is forwarded
Receive Only	Receiving	Encrypted traffic matching policy is decrypted Encrypted traffic not matching policy is dropped Clear traffic matching policy is forwarded Clear traffic not matching policy is forwarded
	Sending	Clear traffic matching policy is **forwarded** Clear traffic not matching policy is forwarded
Passive Mode	Receiving	Encrypted traffic matching policy is decrypted Encrypted traffic not matching policy is dropped Clear traffic matching policy is forwarded Clear traffic not matching policy is forwarded
	Sending	Clear traffic matching policy is forwarded Clear traffic not matching policy is forwarded

4.4.4 Migration Strategy: Receive-Only SAs

Migration to a GET VPN network from a network without encryption, receive-only SAs can be used while encryption services are deployed. To change IPsec SA behavior, issue the following command on the KS:

```
crypto gdoi group dgvpn1
 server local
  sa receive-only
!
```

IPsec SAs are now installed in an inbound only direction. This can be verified by issuing the following command on the GM (or KS):


```
GM#sh cry gdoi

GROUP INFORMATION

    Group Name                  : dgvpn1
    Group Identity              : 101
    Rekeys received             : 4
    IPSec SA Direction          : Inbound Only
```

In `Inbound Only` mode, the IPsec SA on the GM can receive both encrypted and cleartext traffic. The outgoing GM traffic is not encrypted and is sent in clear.

Configuring `Inbound Only` mode enables GET VPN to be deployed using the phased approach, a few devices at a time. Devices that have been configured as GET VPN GMs have the control plane ready, but the data plane is not engaged; no encryption will occur on any of the devices. This enables the GMs to be able to talk to other CEs that have not been configured as GMs.

When the control plane (registration, rekeys, SA installation, and so on) operation of GET VPN is verified, encryption can be turned on all devices at the same time by removing the **`sa receive-only`** command from the KS. This results in an immediate rekey being sent from the KS, changing the direction of the SA on the GMs to “Inbound Optional”.

```
KS1(config)#crypto gdoi group dgvpn1
KS1(config-gdoi-group)# server local
KS1(gdoi-local-server)#no sa receive-only
KS1(gdoi-local-server)#end
KS1#
May 11 23:39:03.442: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for
group dgvpn1 from address 3.3.3.3 with seq # 1
```

```
GM#sh cry gdoi

GROUP INFORMATION

    Group Name                  : dgvpn1
    Group Identity              : 101
    Rekeys received             : 5
    IPSec SA Direction          : Inbound Optional
```

In `Inbound Optional` mode, the IPsec SA on the GM can receive encrypted and plain text traffic. Outgoing GM traffic is encrypted (unlike `Inbound Only` mode). The GM actually automatically transitions from Receive-Only through Passive-Mode to Conformant-Mode.

Subsequent rekeys change the IPsec SA to normal mode, where SAs can receive only encrypted traffic and always send encrypted traffic:

```
GM#sh cry gdoi

GROUP INFORMATION

    Group Name                  : dgvpn1
    Group Identity              : 101
    Rekeys received             : 0
    IPSec SA Direction          : Both
```


4.4.5 Migration Strategy – Testing before Deploying

GET VPN configuration enables a user to override KS configuration and change the direction of the IPsec SAs. This functionality can be very useful when testing encryption on certain sites or parts of the network before deploying encryption over the complete network.

For example, if a GET VPN deployment is completed with KS configured in "receive-only" mode, a user can change the direction of the IPsec SA on a few sites to test encryption and application behavior on those sites without affecting the complete network. The following CLI changes the IPsec SA direction:

```
GM#crypto gdoi gm ipsec direction ?

  both      IPsec SA will only accept cipher text and will encrypt the packet
            before forwarding it out
  inbound   Specify IPsec SA inbound options

GM#crypto gdoi gm ipsec direction inbound ?

  only       IPsec SA will accept both cipher/plain text and will forward the
             packet in clear.
  optional   IPsec SA will accept both cipher/plain text and will encrypt the
             packet before forwarding it out
```

Note: Rekey from the KS overrides user-defined IPsec SA direction settings to the KS configured value. Users must to reissue the command to get the desired setting.

5 Provisioning, Verification, and Monitoring

This chapter describes using Cisco Security Manager (CSM) to deploy GET VPN, the syslog capabilities of GET VPN, GET VPN verification, and Simple Network Management Protocol (SNMP) polling.

5.1 Deploying GET VPN using CSM

CSM 3.2 or below do not support GET VPN deployment via Site to Site VPN Manager. GET VPN support via VPN Manger will be available in future releases of CSM. Currently, the powerful “Flex Configuration” feature of the CSM can be used for GET VPN deployments.

Configuration commands can be used to create FlexConfig policy objects in the FlexConfig Editor, with or without additional scripting language instructions.

Note: FlexConfig provisioning is similar to deploying an IOS configuration template to one or more devices. Modifying and redeploying some configurations might require negating certain previously deployed commands using the **`no`** keyword.

GET VPN can be deployed using FlexConfig by following a few easy steps:

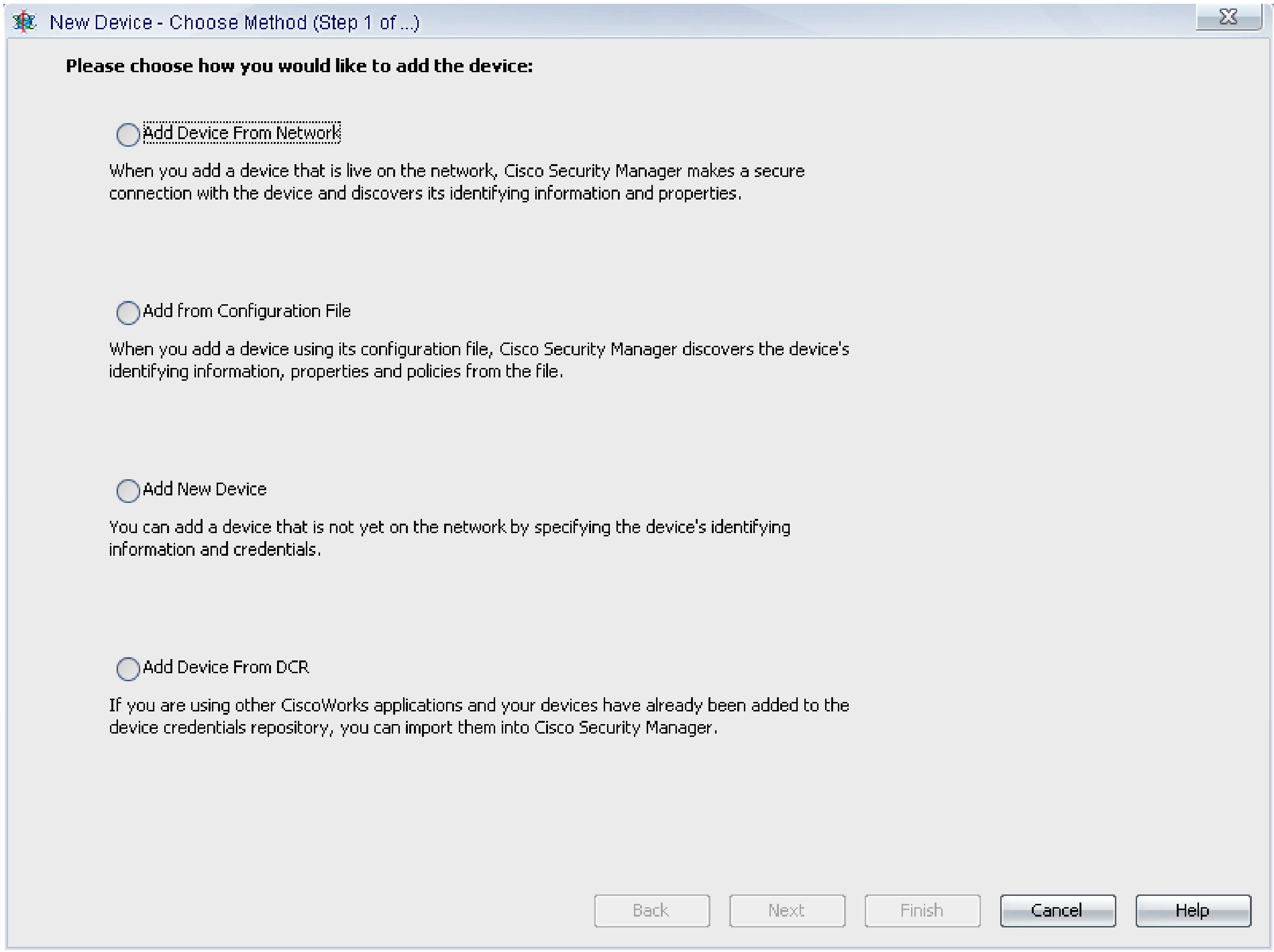
Step 1. Configure IOS Device to accept CSM connections:

- Set up basic routing for connectivity between CSM and IOS devices.
- Set up HTTPS: **`ip http secureserver`** configuration in the IOS devices (this results in the generation of a self signed certificate on the router).
- Set up a user with privilege 15 for CSM to connect and configure the device: **`username lab priv 15 password lab`**.
- Set up an enable password on IOS device (not needed if the user is defined with privilege level 15 as shown in last step).

Step 2. Discover devices:

- Use any of the four methods provided in CSM to add the devices
- Follow the prompts and enter the required information (IP addresses, usernames, enable passwords)
- CSM populates the device table

Figure 5-1 CSM Device Discovery



5.1.1 Deploying Key Server Configuration Using CSM

Deploying a key server (KS) configuration with CSM is a little tricky and requires some user intervention. KS configuration requires the following steps:

- Generating and exchanging the RSA keys between the COOP KS
- Configuring pre-shared keys (PSKs) or registering to a certificate authority (CA)
- Defining IKE Phase 1 parameters
- Defining GDOI group and policies
- Defining COOP servers

Generating and exchanging RSA keys is a one-time event and is most easily configured using IOS CLI. Similarly, registering to a CA is an interactive process and should be handled by CLI. Parameters such as policies, group information, and COOP can be deployed either using CSM or CLI and therefore it is important to analyze the KS configuration to decide best way to deploy this configuration.

Consider the following typical KS configuration:


```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set 3des_sha esp-3des esp-sha-hmac
!
crypto ipsec profile gdoi
 set security-association lifetime seconds 7200
 set transform-set 3des_sha
!
crypto gdoi group dgvpn1
 identity number 101
 server local
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa getkey
  rekey transport unicast
  sa ipsec 1
   profile gdoi
   match address ipv4 111
   replay time window-size 2
  address ipv4 7.7.7.7
  redundancy
   local priority 100
   peer address ipv4 3.3.3.3
!
access-list 111 permit ip 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
!
```

In the preceding configuration, while most of the configuration is common between the two KSs, parts of configuration (shown in **bold)** are specific to a particular KS, that is, each KS has its own address, priority, and COOP peers. There are two approaches for the KS configuration deployment:

- Deploy complete KS configuration as a FlexConfig.

  This means the preceding configuration is customized for each KS and is defined in multiple FlexConfigs. These FlexConfigs are then assigned to their respective KSs.

  Any changes to the GET VPN policy (timers, ACL) must be incorporated in every KS FlexConfig before redeploying the configuration.

- Deploy a common KS configuration using shared FlexConfig and use CLI to configure KS specific commands.

  Because any changes made to the shared Flex Configuration must be deployed to all devices sharing the configuration, the chances of inconsistency between KS policies are reduced using this deployment method. This deployment method also helps overcome the inherent issue of configuration sync between the KS in current GET VPN phases.

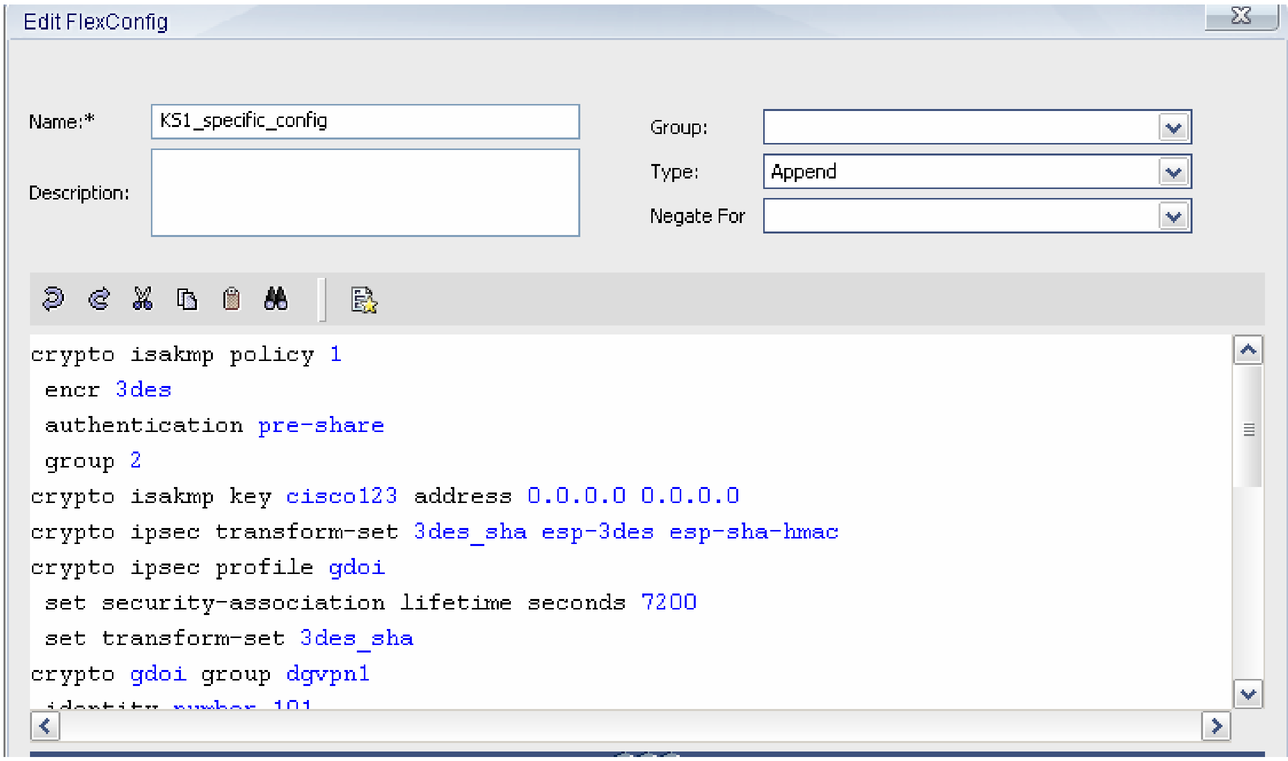
Descriptions of both methods follow. Users can select either method.

5.1.1.1 Adding a KS Specific FlexConfig

1. Click the KS device icon.
2. Click **FlexConfigs**.

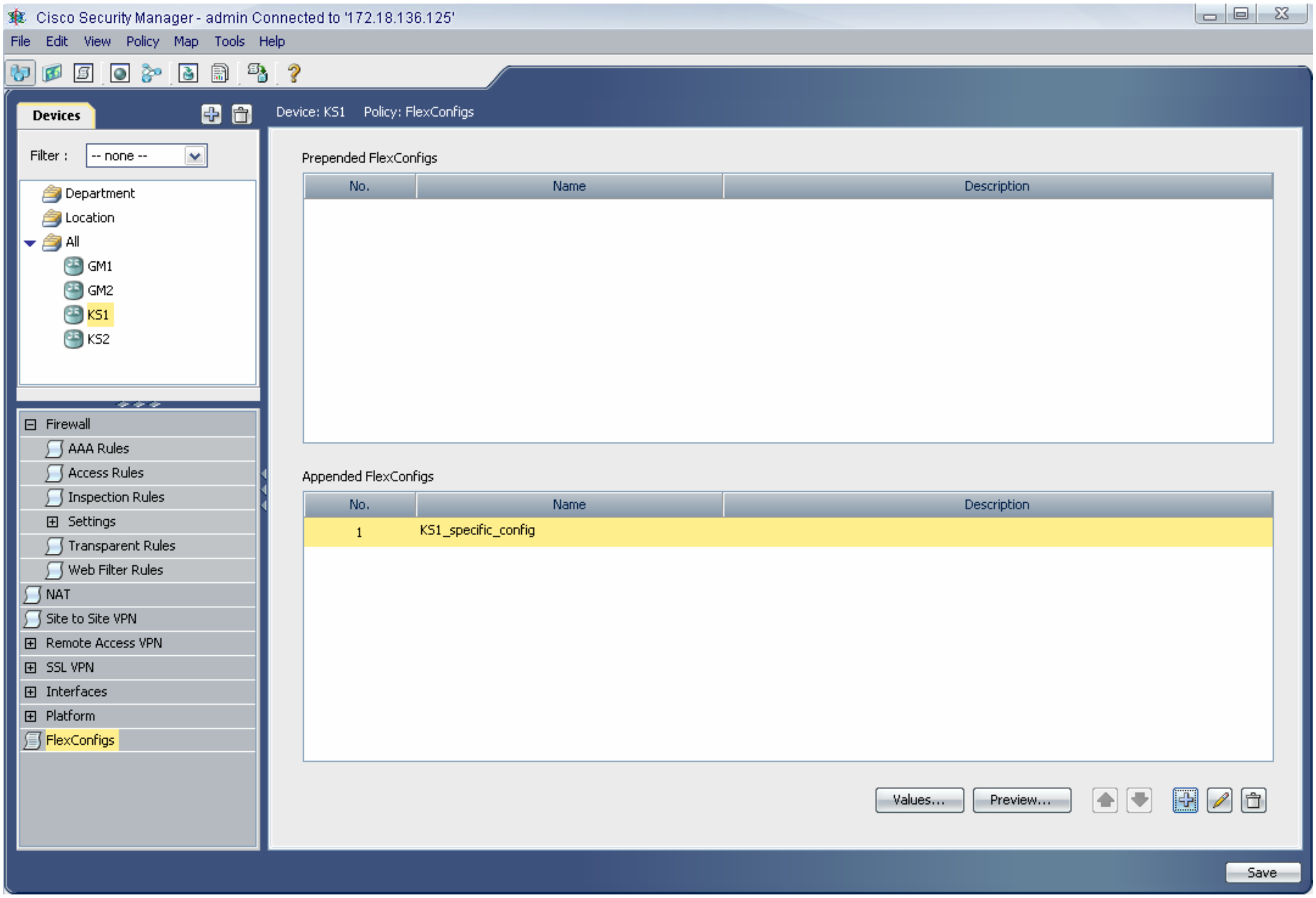
3. Click the + button to add a new FlexConfig.
4. In the **FlexConfig** selector menu box, click + to add a new FlexConfig. Give this configuration a descriptive name, such as **KS1_specific_config**.
5. Add the **complete** KS configuration to the text box.

Figure 5-2 Flex Configuration



6. Click **OK** to save the configuration
7. Double click the configuration (or add it to the right plane)
8. Click **OK**.
9. The new FlexConfig is added to the KS device.

Figure 5-3 Applying Flex Configuration to a Device



10. Follow this procedure to create and attach the FlexConfigs to all KSs
11. Deploy the configurations (**File** >**Submit and Deploy**) to all KSs

5.1.1.2 Adding a Shared FlexConfig

1. Using CLI, add the KS specific configuration on all the KSs in the network:

```
crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0
crypto isakmp key coopkey address 7.7.7.7
!
crypto gdoi group dgvpn1
 identity number 101
 server local
  address ipv4 3.3.3.3
  redundancy
   local priority 100
   peer address ipv4 7.7.7.7
!
```

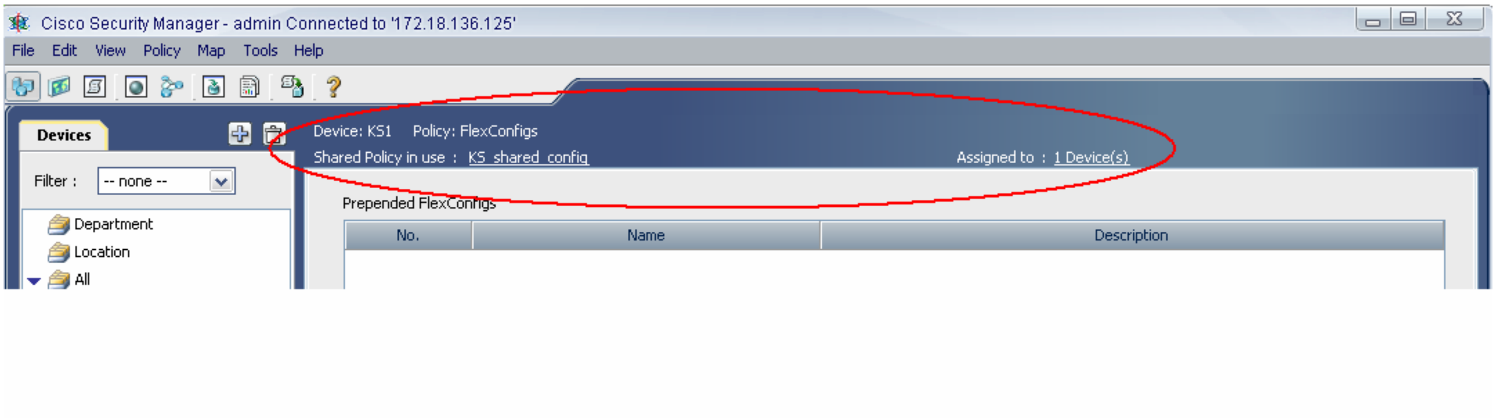
For CSM deployment:

1. Click the KS device icon
2. Click **FlexConfigs**
3. Click the **+** button to add a new FlexConfig
4. In the **FlexConfig** selector menu box, click **+** to add a new FlexConfig. Give this configuration a descriptive name, for example, **KS_shared_config**.
5. Add the shared KS configuration to the text box (similar to the following):

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto ipsec transform-set 3des_sha esp-3des esp-sha-hmac
!
crypto ipsec profile gdoi
 set security-association lifetime seconds 7200
 set transform-set 3des_sha
!
crypto gdoi group dgvpn1
 server local
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa getkey
  rekey transport unicast
  sa ipsec 1
   profile gdoi
   match address ipv4 111
   replay time window-size 2
!
access-list 111 permit ip 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
```

6. Click **OK** to save the configuration
7. Double click the configuration (or add it to the right plane)
8. Click **OK**.
9. The newly created FlexConfig is added to the KS device
10. Click the FlexConfig name to highlight it
11. Go to **Policy** > **Share Policy**.
12. Provide a descriptive name, such as **KS_shared_config**.
13. Click **OK**.
14. In CSM main window, you should see something like:

Figure 5-4 CSM Main Window Showing the Shared Policy



15. Click **Assigned to : 1 Devices**.
16. In the **Shared Policy Assignment for KS_shared_policy** window, add the remaining KSs.
17. Click **OK**.
18. The FlexConfig will appear under all the KSs and the **Assigned to:** field will show the correct number of devices to which the FlexConfig is assigned.
19. Submit and deploy the configuration to the KSs.

Note: After a successful CSM deployment, it is a good idea to issue a **`clear crypto gdoi`** on the KSs to force a COOP election. When configuring multicast rekey on the KS, ensure that the multicast infrastructure is already in place and is working.

5.1.2 Deploying Group Member Configuration Using CSM

Deploying the group member (GM) configuration using CSM is relatively simple. GM configuration can be deployed by creating a FlexConfig and then sharing the configuration amongst all the GMs.

1. In CSM, click any GM device icon
2. Click **FlexConfigs**
3. Click the **+** button to add a new FlexConfig
4. In the FlexConfig selector menu box, click **+** to add a new FlexConfig. Give this configuration a descriptive name, such as **GM_shared_config**.
5. Add the configuration to the text box (similar to following):

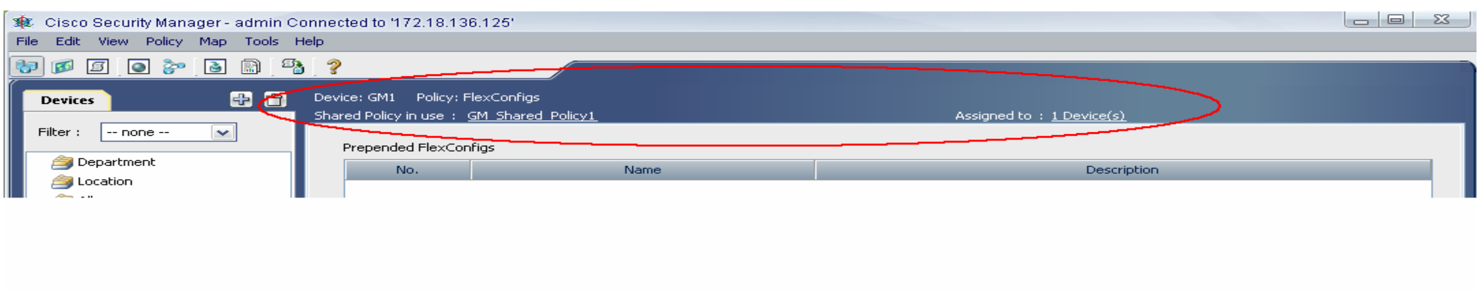
```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
!
crypto isakmp key cisco123 address 3.3.3.3
crypto isakmp key cisco456 address 7.7.7.7
!
crypto gdoi group dgvpn1
 identity number 101
```



```
 server address ipv4 3.3.3.3
 server address ipv4 7.7.7.7
!
!
crypto map getvpn 10 gdoi
 set group dgvpn1
!
interface GigabitEthernet0/0
 crypto map getvpn
```

6. Click **OK** to save the configuration.
7. Double click the configuration (or add it to the right plane)
8. Click **OK**.
9. The newly created FlexConfig is added to the GM device.
10. Click on the FlexConfig name to highlight it
11. Go to **Policy** > **Share Policy**.
12. Provide a descriptive name, such as **GM_shared_policy1**.
13. Click **OK**.
14. In the CSM main window, you should see something like:

Figure 5-5 CSM Main Window Showing the Shared Policy



15. Click **Assigned to : 1 Devices**.
16. In the **Shared Policy Assignment for GM_shared_policy window**, add all the GMs
17. Click **OK**
18. The FlexConfig will appear under all the GMs and the **Assigned to:** field will show the correct number of devices to which the FlexConfig is assigned.
19. Submit and deploy the configuration to the GMs.

If the GMs use different KSs as preferred KSs to distribute the registration load, more than one shared policy must be created and shared. For example:

GM-1 registers to the KSs in the following order:

```
server address ipv4 7.7.7.7 (preferred)
server address ipv4 3.3.3.3
```

But GM-2 registers to the KSs in the reverse order


```
        server address ipv4 3.3.3.3 (preferred)
        server address ipv4 7.7.7.7
```

Two different policies would have to be defined and shared among the appropriate GMs.

5.2 GET VPN Syslog Capabilities

Syslog messages provide a valuable resource for monitoring and troubleshooting GET VPN. To enable syslog on a GM/KS, use the following configuration.

```
!
logging trap <severity level 0 -7>
logging source-interface <interface-type> <interface-number>
logging <syslog server IP address>
!
```

The **logging trap** command can be used to filter syslog messages based on the level. A trap level of 7 means that the router will send all messages to the syslog server.

To enable viewing syslog messages on the CLI, enable terminal monitoring. This command should be used carefully, as a high message rate can overwhelm the CLI.

```
!
terminal monitor
!
```

When using syslog messages, it is recommended to enable NTP on the router.

Table 5-1 lists and describes some of the most common syslog messages associated with GET VPN. For a complete list of syslog messages, refer to following URL:

http://www.cisco.com/en/US/docs/ios/security/configuration/guide/sec_encrypt_trns_vpn.html

Table 5-1: GET VPN Syslog Messages

Syslog Messag	Description
COOP_CONFIG_MISMATCH	The configurations between the primary KS and secondary KS are mismatched.
COOP_KS_ELECTION	The local KS has entered the election process in a group.
COOP_KS_REACH	The reachability between configured cooperative (COOP) KSs is restored.
COOP_KS_TRANS_TO_PRI	The local KS transitioned to a primary role from being a secondary server in a group.
COOP_KS_UNAUTH	An authorized remote server tried to contact the local KS in a group; this c ould be considered a hostile event.
COOP_KS_UNREACH	The reachability between the configured cooperative KSs is lost; this could be considered a hostile event.
GDOI_ACL_NUM	The ACL has too many entries. GDOI honors only the first 100 ACL entries.
GM_CLEAR_REGISTER	The **clear crypto gdoi** command was executed by the local GM.
GM_CM_ATTACH	A crypto map was attached for the local GM.
GM_CM_DETACH	A crypto map was detached for the local GM.
GM_ENABLE_GDOI_CM	GM has enabled ACL on a GDOI crypto map in a group with a KS.

GM_INCOMPLETE_CFG	Registration cannot be completed because the GDOI group configuration might be missing the group ID, server ID, or both.
GM_NO_IPSEC_FLOWS	The hardware limitation for IPsec flow limit reached. Cannot create any more IPsec SAs.
GM_RE_REGISTER	IPsec SA created for one group may have been expired or cleared. Need to reregister to the KS.
GM_RECV_REKEY	Rekey received.
GM_REGS_COMPL	Registration complete.
GM_REKEY_TRANS_2_MULTI	GM has transitioned from using a unicast rekey mechanism to using a multicast mechanism.
GM_REKEY_TRANS_2_UNI	GM has transitioned from using a multicast rekey mechanism to using a unicast mechanism.
GM_UNREGISTER	A GM has left the group.
KS_GM_REVOKED	During rekey protocol, an unauthorized member tried to join a group. Could be considered a hostile event.
KS_REKEY_TRANS_2_MULTI	Group has transitioned from using a unicast rekey mechanism to a multicast mechanism.
KS_REKEY_TRANS_2_UNI	Group has transitioned from using a multicast rekey mechanism to using a unicast mechanism.
KS_SEND_MCAST_REKEY	Sending multicast rekey.
KS_SEND_UNICAST_REKEY	Sending unicast rekey.
KS_UNAUTHORIZED	During GDOI registration protocol, an unauthorized member tried to join a group. Could be considered a hostile event.
PSEUDO_TIME_LARGE	A GM has received a pseudotime with a value that is largely different from its own pseudotime.
REPLAY_FAILED	A GM or KS has failed an anti-replay check.
UNAUTHORIZED_IDENTITY	The registration request was dropped because the requesting device was not authorized to join the group.
UNAUTHORIZED_IPADDR	The registration request was dropped because the requesting device was not authorized to join the group.

5.3 GET VPN Verification

GET VPN provides an enhanced set of syslog and show commands to verify the normal functionality and debug network problems. Some of the commonly used syslog and show commands are shown below. Because error and failure conditions are common in networks, these show commands and syslog messages can help troubleshoot failure scenarios.

5.3.1 Verifying KS Operation

This section describes various show commands that can be used to verify KS operation.

5.3.1.1 show crypto gdoi

To display information about GET VPN configuration on the KS, use the **show crypto gdoi** command in privileged EXEC mode.

```
Primary_Key_Server#sh crypto gdoi
GROUP INFORMATION

    Group Name                : getvpn (Unicast)
    Group Identity            : 1234
    Group Members             : 2
    IPSec SA Direction        : Both
    Active Group Server       : Local
    Redundancy                : Configured
        Local Address         : 100.1.1.1
        Local Priority        : 100
        Local KS Status       : Alive
        Local KS Role         : Primary
    Group Rekey Lifetime      : 86400 secs
    Group Rekey
        Remaining Lifetime    : 85505 secs
    Rekey Retransmit Period   : 10 secs
    Rekey Retransmit Attempts: 2
    Group Retransmit
        Remaining Lifetime    : 0 secs

      IPSec SA Number         : 1
      IPSec SA Rekey Lifetime: 7200 secs
      Profile Name            : gdoi-profile-getvpn
      Replay method           : Time Based
      Replay Window Size      : 5
      SA Rekey
         Remaining Lifetime   : 789 secs
      ACL Configured          : access-list 199

    Group Server list         : Local
```

5.3.1.2 show crypto gdoi ks acl

The **show crypto gdoi** command also shows the ACL 199 is being used to define interesting traffic (traffic to be encrypted or not-encrypted) for the group. To verify the ACL policies on the KS, use the **show crypto gdoi ks acl** command:

```
Primary_Key_Server#show crypto gdoi ks acl
Group Name: dgvpn1
 Configured ACL:
   access-list 199 deny igmp any any
   access-list 199 deny pim any any
   access-list 199 deny eigrp any any
   access-list 199 deny udp any any port = 500
   access-list 199 deny udp any any port = 848
   access-list 199 permit ip 151.0.0.0 0.255.255.255 151.0.0.0 0.255.255.255
   access-list 199 deny ip any any
```

5.3.1.3 show crypto gdoi ks rekey

The **`show crypto gdoi ks rekey`** command provides information about the rekey statistics, such as. how many rekeys were sent out, how many times rekey was retransmitted, the configured lifetime and remaining lifetime and so on.

5.3.1.3.1 Multicast Rekey

```
Primary_Key_Server#sh crypto gdoi ks rekey
Group getvpn (Multicast)
    Number of Rekeys sent                : 2
    Number of Rekeys retransmitted       : 4
    KEK rekey lifetime (sec)             : 86400
        Remaining lifetime (sec)         : 84664
    Retransmit period                    : 10
    Number of retransmissions            : 2
    IPSec SA 1  lifetime (sec)           : 7200
        Remaining lifetime (sec)         : 745
    Number of registrations after rekey : 0
    Multicast destination address        : 239.77.77.77
```

5.3.1.3.2 Unicast Rekey

```
Primary_Key_Server#sh crypto gdoi ks rekey
Group getvpn (Unicast)
    Number of Rekeys sent                : 1
    Number of Rekeys retransmitted       : 0
    KEK rekey lifetime (sec)             : 86400
        Remaining lifetime (sec)         : 85597
    Retransmit period                    : 10
    Number of retransmissions            : 2
    IPSec SA 1  lifetime (sec)           : 7200
        Remaining lifetime (sec)         : 883
```

5.3.1.4 show crypto isakmp sa

To display all current Internet Key Exchange (IKE) security associations (SAs), use the **`show crypto isakmp sa`** command in EXEC mode. Note that the IKE session between the GM and KS (shown as GDOI_IDLE) will time out after the configured IKE lifetime. An IKE session is only required for initial registration and does not need to stay up for normal GET VPN operation. A rekey SA (shown as GDOI_REKEY) however always stays in the IKE database.

```
Primary_Key_Server#sh crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
100.1.1.1       100.1.1.5       GDOI_IDLE         1002    0 ACTIVE
100.1.1.1       100.1.1.9       GDOI_IDLE         1003    0 ACTIVE
100.1.1.1       100.1.1.13      GDOI_IDLE         1004    0 ACTIVE
100.1.1.13      100.1.1.1       GDOI_REKEY           0    0 ACTIVE
```

5.3.1.5 show crypto isakmp sa detail

To display detailed information about all current Internet Key Exchange (IKE) security associations (SAs), use the **`show crypto isakmp sa detail`** command in EXEC mode. This command provides details about time left on the SA, encryption engine in use as well as details of authentication method etc.

5.3.1.5.1 PSKs

```
Primary_Key_Server#sh crypto isakmp sa detail
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id  Local           Remote          I-VRF    Status Encr Hash Auth DH Lifetime Cap.

1002  100.1.1.1       100.1.1.5                ACTIVE aes  sha  psk  2  23:55:17
       Engine-id:Conn-id =  SW:2

1003  100.1.1.1       100.1.1.9                ACTIVE aes  sha  psk  2  23:55:31
       Engine-id:Conn-id =  SW:3

1004  100.1.1.1       100.1.1.13               ACTIVE aes  sha  psk  2  23:55:43
       Engine-id:Conn-id =  SW:4
```

5.3.1.5.2 PKI

```
Primary_Key_Server#sh crypto isakmp sa detail
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption
IPv4 Crypto ISAKMP SA

C-id  Local           Remote          I-VRF    Status Encr Hash Auth DH Lifetime Cap.

1007  100.1.1.1       100.1.1.5                ACTIVE aes  sha  rsig 2  23:58:12
       Engine-id:Conn-id =  SW:7

1008  100.1.1.1       100.1.1.9                ACTIVE aes sha  rsig 2  23:58:22
       Engine-id:Conn-id =  SW:8

1009  100.1.1.1       100.1.1.13               ACTIVE aes sha  rsig 2  23:58:32
```

5.3.1.6 show crypto gdoi ks member

To display information about GMs registered with a Server, use the **sh crypto gdoi ks member** command. This command can be executed on any of the COOP KS. The output also indicates the KS address with which the GM originally registered.

```
Primary_Key_Server#sh crypto gdoi ks member

Group Member Information :

Number of rekeys sent for group getvpn : 1

Group Member ID   : 100.1.1.9
Group ID          : 1234
Group Name        : getvpn
Key Server ID     : 100.1.1.1
Rekeys sent       : 1
Rekeys retries    : 0
Rekey Acks Rcvd   : 1
Rekey Acks missed : 0

Sent seq num :    1    0    0    0
Rcvd seq num :    1    0    0    0
```



```
Group Member ID    : 100.1.1.13
Group ID           : 1234
Group Name         : getvpn
Key Server ID      : 100.1.1.1
Rekeys sent        : 1
Rekeys retries     : 0
Rekey Acks Rcvd    : 1
Rekey Acks missed  : 0

Sent seq num :    1    0    0    0
Rcvd seq num :    1    0    0    0
```

The **show crypto gdoi ks member** command also shows detailed information about number of rekeys and rekey retries sent to a specific GM. An incrementing count of 'rekeys retries' and 'Rekey Acks missed' can indicate problems with a GM.

Shown here is a sample output for a GM 138.1.1.1 that is currently not responding to rekeys. This is indicated by missing rekey Acks for sequence numbers 13 and 14.

```
Primary_Key_Server#show crypto gdoi ks member | b 138.1.1.1
Group Member ID    : 138.1.1.1
Group ID           : 61440
Group Name         : dgvpn1
Key Server ID      : 139.1.1.1
Rekeys sent        : 8
Rekeys retries     : 6
Rekey Acks Rcvd    : 5
Rekey Acks missed  : 2

Sent seq num :    13    14    0    0
Rcvd seq num :    0    0    0    0
```

5.3.1.7 show crypto gdoi ks coop

To display information about COOP KSs in a GET VPN environment, use the **sh crypto gdoi ks coop** command. This command provides detailed information about the priorities and roles of various COOP devices.

```
Primary_Key_Server#sh crypto gdoi ks coop
Crypto Gdoi Group Name :getvpn
        Group handle: 2147483652, Local Key Server handle: 2147483652

        Local Address: 100.1.1.1
        Local Priority: 100
        Local KS Role: Primary   , Local KS Status: Alive
        Primary Timers:
                Primary Refresh Policy Time: 20
                Remaining Time: 17
                Antireplay Sequence Number: 64

        Peer Sessions:
        Session 1:
                Server handle: 2147483653
                Peer Address: 100.1.1.5
                Peer Priority: 75
                Peer KS Role: Secondary , Peer KS Status: Alive
                Antireplay Sequence Number: 2
```



```
                IKE status: Established
                Counters:
                    Ann msgs sent: 61
                    Ann msgs sent with reply request: 1
                    Ann msgs recv: 1
                    Ann msgs recv with reply request: 2
                    Packet sent drops: 2
                    Packet Recv drops: 0
                    Total bytes sent: 33402
                    Total bytes recv: 1170
```

```
COOP_Key_Server#sh crypto gdoi ks coop
Crypto Gdoi Group Name :getvpn
        Group handle: 2147483650, Local Key Server handle: 2147483650

        Local Address: 100.1.1.5
        Local Priority: 75
        Local KS Role: Secondary , Local KS Status: Alive
        Secondary Timers:
                Sec Primary Periodic Time: 30
                Remaining Time: 11, Retries: 0
                Antireplay Sequence Number: 3

        Peer Sessions:
        Session 1:
                Server handle: 2147483651
                Peer Address: 100.1.1.1
                Peer Priority: 100
                Peer KS Role: Primary   , Peer KS Status: Alive
                Antireplay Sequence Number: 63

                IKE status: Established
                Counters:
                    Ann msgs sent: 0
                    Ann msgs sent with reply request: 2
                    Ann msgs recv: 61
                    Ann msgs recv with reply request: 0
                    Packet sent drops: 1
                    Packet Recv drops: 0
                    Total bytes sent: 1074
                    Total bytes recv: 33346
```

5.3.2 Verifying GM Operation

This section describes various show commands that can be used to verify GM operation.

5.3.2.1 show crypto isakmp sa

To display all current IKE SAs, use the **show crypto isakmp sa** command. Note that the IKE session between the GM (shown as GDOI_IDLE) and KS will time out after the configured time. When a GM registers with the KS, two IKE SAs are created - GDOI_IDLE and GDOI_REKEY. GDOI_IDLE SA gets deleted after the configured lifetime but GDOI_REKEY SA stays in the IKE database.

```
GroupMember-1#sh crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
```



```
100.1.1.9        100.1.1.1        GDOI_REKEY        2060    0 ACTIVE
100.1.1.1        100.1.1.9        GDOI_IDLE         4038    0 ACTIVE
```

5.3.2.2 show crypto gdoi

To display information about GET VPN configuration on the GM, use the **show crypto gdoi** command on the GM. This command provides a summary of all the useful GET VPN parameters.

```
GroupMember-1#sh crypto gdoi
GROUP INFORMATION

    Group Name               : getvpn
    Group Identity           : 1234
    Rekeys received          : 1
    IPSec SA Direction       : Both
    Active Group Server      : 100.1.1.1
    Group Server list        : 100.1.1.1
                               100.1.1.5

    GM Reregisters in        : 618 secs
    Rekey Received(hh:mm:ss) : 00:03:37


    Rekeys received
         Cumulative          : 1
         After registration  : 1
    Rekey Acks sent          : 1

 ACL Downloaded From KS 100.1.1.1:
   access-list  permit ip 10.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255

KEK POLICY:
    Rekey Transport Type     : Unicast
    Lifetime (secs)          : 85615
    Encrypt Algorithm        : 3DES
    Key Size                 : 192
    Sig Hash Algorithm       : HMAC_AUTH_SHA
    Sig Key Length (bits)    : 1024

TEK POLICY:
  FastEthernet0/1:
    IPsec SA:
        sa direction:inbound
        spi: 0xC920B5B0(3374364080)
        transform: esp-aes esp-sha-hmac
        sa timing:remaining key lifetime (sec): (681)
        Anti-Replay(Time Based) : 5 sec interval

    IPsec SA:
        sa direction:outbound
        spi: 0xC920B5B0(3374364080)
        transform: esp-aes esp-sha-hmac
        sa timing:remaining key lifetime (sec): (681)
        Anti-Replay(Time Based) : 5 sec interval
```


5.3.2.3 show crypto gdoi gm rekey

The **show crypto gdoi gm rekey** command provides rekey statistics, for example, how many rekeys were received since the first registration, how many rekeys were received since the most recent registration, how many ACKs were sent (unicast rekey), multicast group (multicast rekey), and so on.

5.3.2.3.1 Multicast Rekey

```
GroupMember-1#sh crypto gdoi gm rekey
Group getvpn (Multicast)
    Number of Rekeys received (cumulative)       : 9
    Number of Rekeys received after registration : 9
    Multicast destination address                : 239.77.77.77

Rekey (KEK) SA information :
          dst             src             conn-id  my-cookie  his-cookie
New     : 239.77.77.77    100.1.1.1         2063   3E281825   0F770714
Current : 239.77.77.77    100.1.1.1         2063   3E281825   0F770714
Previous: ---             ---               ---    ---        ---

GroupMember-1#sh ip igmp groups

IGMP Connected Group Membership
Group Address    Interface               Uptime    Expires   LastReporter
239.77.77.77     FastEthernet0/1         3w1d      00:02:12  100.1.1.9
224.0.1.40       FastEthernet0/1         3w1d      00:02:10  100.1.1.9
```

5.3.2.3.2 Unicast Rekey

```
GroupMember-1#sh crypto gdoi gm rekey
Group getvpn (Unicast)
    Number of Rekeys received (cumulative)       : 1
    Number of Rekeys received after registration : 1
    Number of Rekey Acks sent                    : 1

Rekey (KEK) SA information :
          dst             src             conn-id  my-cookie  his-cookie
New     : 100.1.1.9       100.1.1.1         2065   CEC03E80   F3D7CB96
Current : 100.1.1.9       100.1.1.1         2065   CEC03E80   F3D7CB96
Previous: ---             ---               ---    ---        ---
```

5.3.3 KS and GM Syslog Messages

This section describes various syslog messages that can be used to verify GET VPN operation.

5.3.3.1 Syslog GM_REGSTER and GM_REGS_COMPL

When a GM first comes up, it goes through the registration process to the first listed KS in its configuration. The **GM_REGSTER** syslog message indicates that the GM started the registration process.

```
%CRYPTO-5-GM_REGSTER: Start registration to KS 139.1.1.1 for group dgvpn
using address 132.102.1.1
```


The **GM_REGS_COMPL** message indicates that the GM successfully completed registration. This message also identifies the KS with which the GM registered.

```
%GDOI-5-GM_REGS_COMPL: Registration to KS 139.1.1.1 complete for group
dgvpn using address 132.102.1.1
```

5.3.3.2 KS_SEND_<UNI | MULTI>CAST_REKEY:

The **KS_SEND_UNICAST_REKEY** and **KS_SEND_MULTICAST_REKEY** syslog messages are seen when the primary KS sends out a rekey to the group. These messages can be used to verify rekey behavior (timing, retransmits etc) in the GET VPN network. The message below is an example of a unicast rekey message.

```
%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group getvpn from
address 100.1.1.1 with seq # 1
```

In case of unicast rekey mechanism, the rekey message can be an indicator of GM or network failure. If all GMs in the GET VPN group reply back to a unicast rekey, rekey syslog messages are displayed with consecutive incrementing sequence numbers.

```
%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group getvpn from
address 100.1.1.1 with seq # 1
%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group getvpn from
address 100.1.1.1 with seq # 2
%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group getvpn from
address 100.1.1.1 with seq # 3
```

However, if one or more GMs do not reply back with a rekey ACK, the primary KS sends out additional rekey messages (retransmissions) to the failing GMs. These retransmissions increment the rekey sequence number abnormally (based on number of retransmits sent). If syslog does not show the rekey sequence numbers incrementing properly (last sequence number + 1), this indicates that the primary KS is sending out some rekey retransmissions because ACKs from some GMs is not being received.

```
%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group getvpn from
address 100.1.1.1 with seq # 9
%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group getvpn from
address 100.1.1.1 with seq # 12
```

5.3.3.3 Syslog GM_RECV_REKEY

The **GM_RECV_REKEY** message at the GM indicates that the GM has received a rekey from the primary KS. In the message shown below, the GM 138.1.1.1 received a rekey with sequence # 5 from the primary KS 139.1.1.1

```
%GDOI-5-GM_RECV_REKEY: Received Rekey for group dgvpn from 139.1.1.1 to
138.1.1.1 with seq # 5
```

5.3.3.4 Syslog GM_RE_REGISTER

The **GM_RE_REGISTER** message indicates that the GM did not receive the rekeys from the KS. GM attempts to reregister with the KS 60 seconds before the expiration of current IPsec SAs.

```
%GDOI-4-GM_RE_REGISTER: The IPSec SA created for group dgvpn may have
expired/been cleared, or didn't go through. Re-register to KS.
%CRYPTO-5-GM_REGSTER: Start registration to KS 139.1.1.1 for group dgvpn
using address 132.102.1.1
```


5.3.3.5 Syslog GM_REGSTER_IF_DOWN

The **GM_REGSTER_IF_DOWN** message indicates that the source interface for the crypto map is down and hence GM is unable to register. If crypto map is sourced directly from a physical interface, this message is seen when that interface is down. If crypto map is sourced from a loopback interface, this message is seen only when the loopback interface is disabled.

```
%CRYPTO-4-GM_REGSTER_IF_DOWN: Can't start GDOI registeration as interface
GigabitEthernet0/1 is down
```

5.3.3.6 Syslog GM_CONN_NEXT_SER

During the registration process, if the GM is not able to reach the first KS in its configuration, it tries the next KS listed in its configuration. The **GM_CONN_NEXT_SER** message is seen every time GM selects the next server in the list to register.

```
%CRYPTO-5-GM_CONN_NEXT_SER: GM is connecting to next key server from the
list
```

5.3.3.7 Syslog GM_DELETE

The **GM_DELETE** syslog message is displayed on all KSs when they delete a GM from the database. This message is only seen in the case of unicast rekeys, and after the GM failed to respond to three consecutive rekeys (and rekey retransmissions).

```
%GDOI-4-GM_DELETE: GM 100.1.1.9 deleted from group getvpn.
```

5.3.4 COOP Syslog Messages

This section describes various syslog messages that can be used to verify COOP operation in a GET VPN environment.

5.3.4.1 Syslog GDOI-5-COOP_KS_ELECTION

When the COOP KSs come up (or GDOI is cleared), they go through COOP election process. At this instance following syslog messages can be seen on all the KSs.

```
%GDOI-5-COOP_KS_ELECTION: KS entering election mode in group getvpn
(Previous Primary = NONE)
```

5.3.4.2 Syslog GDOI-5-COOP_KS_TRANS_TO_PRI

When the COOP election process is completed, **COOP_KS_TRANS_TO_PRI** message displays information about newly elected primary KS. This message is seen on both the primary and the secondary KSs. When the KSs come up for the first time and enter election process, there is no primary KS on the network. The previous primary is shown as **NONE**.

```
%GDOI-5-COOP_KS_TRANS_TO_PRI: KS 100.1.1.1 in group getvpn transitioned to
Primary (Previous Primary = NONE)
```

If a primary KS failure causes a reelection, the syslog message also indicates the IP address of the previous primary.

```
%GDOI-5-COOP_KS_TRANS_TO_PRI: KS 100.1.1.5 in group getvpn transitioned to
Primary (Previous Primary = 100.1.1.1)
```

5.3.4.3 Syslog COOP_KS_UNREACH and COOP_KS_REACH

The **COOP_KS_UNREACH** message is displayed when a KS loses connectivity with peer COOP KSs. A primary KS tracks the state of all secondary KSs and uses this message to indicate loss of connectivity to a secondary KS. A secondary KS only tracks the state of the primary KS. This message on the secondary KS indicates loss of connectivity with the primary KS.

Note: Although the primary KS may lose connectivity to more than 1 secondary KSs, a syslog message is currently only displayed by the primary KS for the first failed Secondary KS (CSCsl52477).

The following message indicates that the KS lost connectivity to COOP KS 100.1.1.5.

```
%GDOI-3-COOP_KS_UNREACH: Cooperative KS 100.1.1.5 Unreachable in group
getvpn
```

When the connectivity is restored among the COOP KSs, a COOP_KS_REACH message is displayed.

```
%GDOI-5-COOP_KS_REACH: Reachability restored with Cooperative KS 100.1.1.5
in group getvpn.
```

5.4 SNMP Polling

SNMP is an application-layer communication protocol that allows network devices to exchange management information. Through SNMP, network administrators can manage network performance, find and solve network problems.

GET VPN currently does not have support GET VPN related SNMP MIB objects. However, SNMP polling can poll other objects which can be used for tracking device performance. SNMP polling tools can be used that periodically poll the interesting objects and many of these tools also provide plotting functionality that allows network administrators to analyze performance data collected for a period of time.

Table 5-2 lists some of the SNMP objects that can be used to monitor GET VPN device performance. The OIDs of the SNMP objects can be obtained using the “SNMP Object Navigator” tool at www.cisco.com

Table 5-2: SNMP Objects Used for SNMP Polling

SNMP Object	Description
cpmCPUTotal1min	The overall CPU busy percentage in the last minute
cpmCPUTotal5min	The overall CPU busy percentage in the last five minutes
ciscoMemoryPoolUsed	The number of bytes from the memory pool currently in use by applications on the managed device
ciscoMemoryPoolFree	The number of bytes from the memory pool that are currently unused on the managed device. Note that the sum of ciscoMemoryPoolUsed and ciscoMemoryPoolFree is the total amount of memory in the pool.
cikeGlobalActiveTunnels	The number of currently active IPsec Phase-1 IKE Tunnels (Useful for KS)
cipSecGlobalActiveTunnels	The total number of currently active IPsec Phase-2 Tunnels (Useful for GMs)

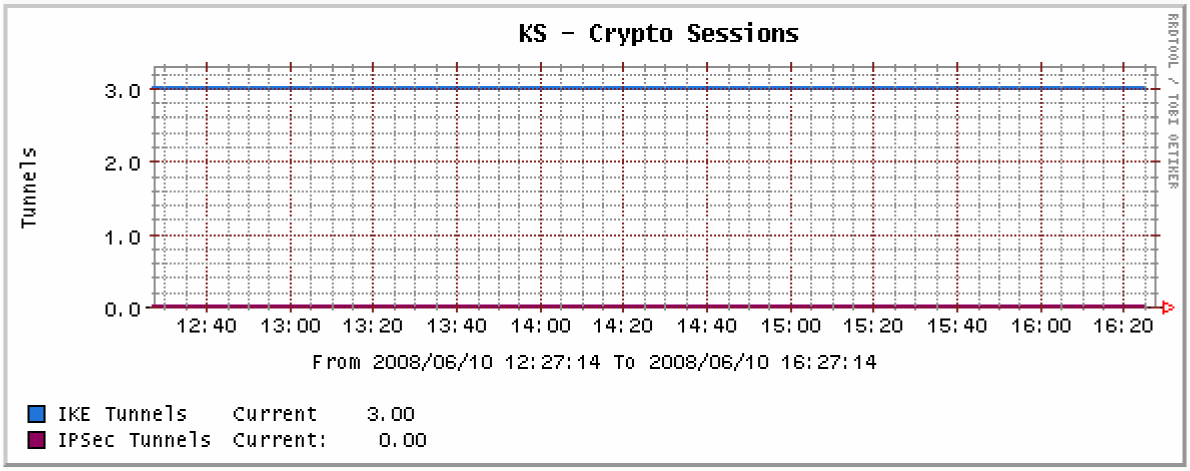
The following sections describe how to use SNMP plots to analyze GET VPN network behavior.

5.4.1 Normal Behavior

In a stable state, a KS maintains active IKE SAs for each COOP KS. For example, if there are four KSs, each KS has three IKE SAs in the GDOI_IDLE state.

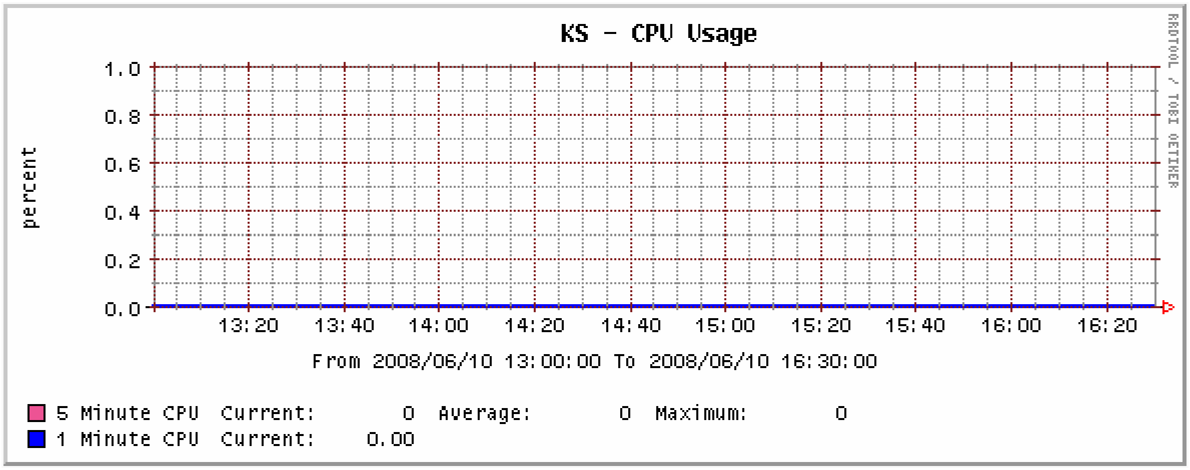
Figure 5-6 shows a sample plot obtained by SNMP polling the KS. In case of four COOP KSs, each KS maintains 3 IKE tunnels.

Figure 5-6: SNMP Polling: Steady State – IKE Tunnels



Because a KS is busy only at registration time or during rekeys, KS CPU use stays close to 0%.

Figure 5-7: SNMP Polling: Steady state - CPU

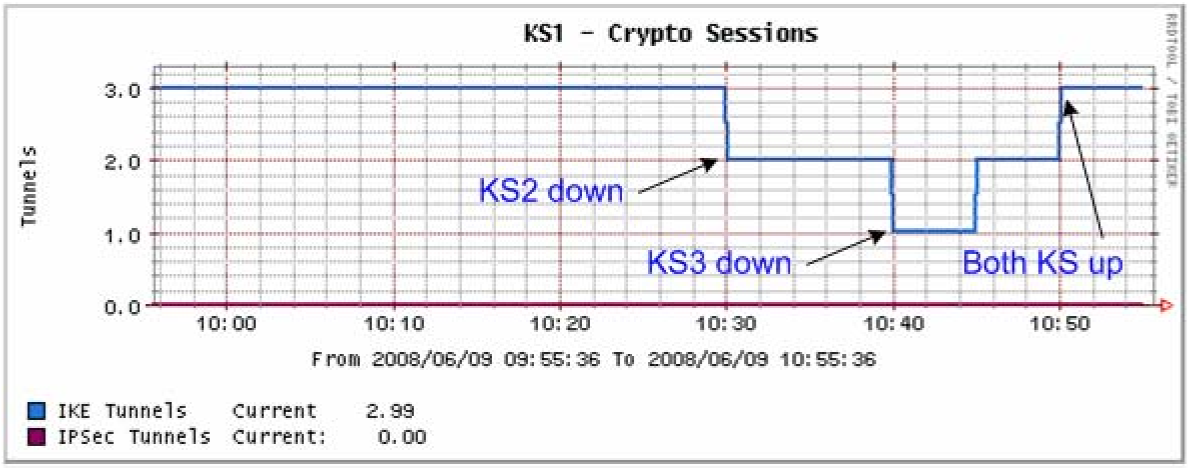


Note: A CPU spike is seen at registration time and during rekey, as shown in Table 3-2.

5.4.2 COOP KS Failure

In case of COOP server failure, the number of SAs goes down, indicating a communication loss with one or more of COOP KSs. Figure 5-8 shows that at time 10:30, an IKE SA expired at KS1, indicating that the KS lost reachability to a COOP servers. At time 10:50, all IKE SAs are reestablished, indicating that all lost KSs are back up. Further troubleshooting using syslog messages and show commands can help troubleshoot the network problems.

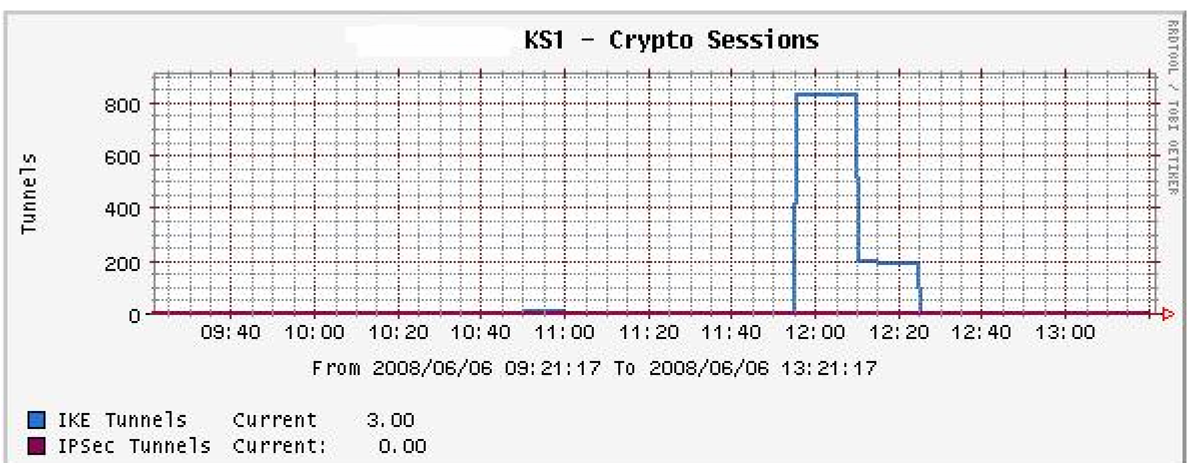
Figure 5-8: SNMP Polling: COOP KS Failure



5.4.3 Registration Burst

If all GMs must reregister at a KS under failure conditions, a burst of IKE sessions can be seen at the KS. Under such circumstances, the SNMP plot for IKE tunnels at the KS look like that in Figure 5-9. The plot shows that at around 12:00 pm, all GMs (more than 800) reregistered to the KS. Because GMs do not need to maintain active IKE sessions with the KS, the IKE sessions were cleared after a configured time.

Figure 5-9: SNMP Polling: Burst of IKE Registrations at a KS



Appendix A Complete Configurations for Section 2

A.1 Using Pre-Shared Keys

A.1.1 Key Server Configuration

```
!
hostname Primary_Key_Server
!
crypto isakmp policy 10
 encr aes
 group 2
 authentication pre-share
crypto isakmp key cisco address 100.1.1.9
crypto isakmp key cisco address 100.1.1.13
crypto isakmp key cisco address 100.1.1.5
!
crypto ipsec transform-set mygdoi-trans esp-aes esp-sha-hmac
!
crypto ipsec profile gdoi-profile-getvpn
 set security-association lifetime seconds 900
 set transform-set mygdoi-trans
!
crypto gdoi group getvpn
 identity number 1234
 server local
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa getvpn-export-general
  rekey transport unicast
  sa ipsec 1
   profile gdoi-profile-getvpn
   match address ipv4 199
   replay time window-size 5
  address ipv4 100.1.1.1
  redundancy
   local priority 100
   peer address ipv4 100.1.1.5
!
interface FastEthernet0/1
 description Outside Interface to PE
 ip address 100.1.1.1 255.255.255.252
 ip nbar protocol-discovery
 ip flow ingress
 load-interval 30
 duplex auto
 speed auto
!
ip route 0.0.0.0 0.0.0.0 100.1.1.2
!
access-list 199 remark ACL Policies to be pushed to GMs
access-list 199 permit ip 10.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255
```

A.1.2 Cooperative Server


```
!
hostname COOP_Key_Server
!
crypto isakmp policy 10
 encr aes
 group 2
 authentication pre-share
crypto isakmp key cisco address 100.1.1.1
crypto isakmp key cisco address 100.1.1.9
crypto isakmp key cisco address 100.1.1.13
!
crypto ipsec transform-set mygdoi-trans esp-aes esp-sha-hmac
!
crypto ipsec profile gdoi-profile-getvpn
 set security-association lifetime seconds 900
 set transform-set mygdoi-trans
!
crypto gdoi group getvpn
 identity number 1234
 server local
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa getvpn-export-general
  rekey transport unicast
  sa ipsec 1
   profile gdoi-profile-getvpn
   match address ipv4 199
   replay time window-size 5
  address ipv4 100.1.1.5
  redundancy
   local priority 75
   peer address ipv4 100.1.1.1
!
interface FastEthernet0/1
 description Outside interface to PE
 ip address 100.1.1.5 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 no negotiation auto
!
ip route 0.0.0.0 0.0.0.0 100.1.1.6
!
access-list 199 remark ACL Policies to be pushed to authenticated group
members
access-list 199 permit ip 10.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255
```

A.1.3 Group Member Configuration

```
!
hostname GroupMember-1
!
crypto isakmp policy 10
 encr aes
 group 2
 authentication pre-share
crypto isakmp key cisco address 100.1.1.1
crypto isakmp key cisco address 100.1.1.5
!
crypto gdoi group getvpn
```



```
 identity number 1234
 server address ipv4 100.1.1.1
 server address ipv4 100.1.1.5
!
crypto map getvpn-map 10 gdoi
 set group getvpn
!
interface FastEthernet0/0
 description Inside interface
 ip address 10.1.11.1 255.255.255.0
!
interface FastEthernet0/1
 description Outside interface to PE
 ip address 100.1.1.9 255.255.255.252
 ip flow ingress
 load-interval 30
 duplex auto
 speed auto
 crypto map getvpn-map
!
router bgp 1111
 no synchronization
 bgp log-neighbor-changes
 network 10.1.11.0 mask 255.255.255.0
 network 100.1.1.8 mask 255.255.255.252
 neighbor 100.1.1.10 remote-as 1000
 no auto-summary
```

A.2 Using Public Key Infrastructure (PKI)

A.2.1 Key Server Configuration

```
!
hostname Primary_Key_Server
!
crypto pki trustpoint GETVPN
 enrollment url http://24.100.100.100:80
 subject-name OU=GETVPN
 revocation-check none
 rsakeypair pkiPKS
!
crypto pki certificate chain GETVPN
 certificate 70
 certificate ca 01
!
crypto isakmp policy 10
 encr aes
 group 2
 authentication rsa-sig
!
crypto ipsec transform-set mygdoi-trans esp-aes esp-sha-hmac
!
crypto ipsec profile gdoi-profile-getvpn
 set security-association lifetime seconds 900
 set transform-set mygdoi-trans
!
crypto gdoi group getvpn
 identity number 1234
```



```
 server local
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa getvpn-export-general
  rekey transport unicast
  sa ipsec 1
   profile gdoi-profile-getvpn
   match address ipv4 199
   replay time window-size 5
  address ipv4 100.1.1.1
  redundancy
   local priority 100
   peer address ipv4 100.1.1.5
!
interface FastEthernet0/1
 description Outside Interface to PE
 ip address 100.1.1.1 255.255.255.252
 ip nbar protocol-discovery
 ip flow ingress
 load-interval 30
 duplex auto
 speed auto
!
ip route 0.0.0.0 0.0.0.0 100.1.1.2
!
access-list 199 remark ACL Policies to be pushed to GMs
access-list 199 permit ip 10.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255
```

A.2.2 Cooperative Server

```
!
hostname COOP_Key_Server
!
crypto pki trustpoint GETVPN
 enrollment url http://24.100.100.100:80
 subject-name OU=GETVPN
 revocation-check none
 rsakeypair pkiCOOP
!
crypto pki certificate chain GETVPN
 certificate 6F
 certificate ca 01
!
crypto isakmp policy 10
 encr aes
 group 2
 authentication rsa-sig
!
crypto ipsec transform-set mygdoi-trans esp-aes esp-sha-hmac
!
crypto ipsec profile gdoi-profile-getvpn
 set security-association lifetime seconds 900
 set transform-set mygdoi-trans
!
crypto gdoi group getvpn
 identity number 1234
 server local
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa getvpn-export-general
  rekey transport unicast
```



```
  sa ipsec 1
   profile gdoi-profile-getvpn
   match address ipv4 199
   replay time window-size 5
  address ipv4 100.1.1.5
  redundancy
   local priority 75
   peer address ipv4 100.1.1.1
!
interface FastEthernet0/1
 description Outside interface to PE
 ip address 100.1.1.5 255.255.255.252
 duplex auto
 speed auto
 media-type rj45
 no negotiation auto
!
ip route 0.0.0.0 0.0.0.0 100.1.1.6
!
access-list 199 remark ACL Policies to be pushed to authenticated group
members
access-list 199 permit ip 10.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255
```

A.2.3 Group Member Configuration

```
!
hostname GroupMember-1
!
crypto pki trustpoint GETVPN
 enrollment url http://24.100.100.100:80
 subject-name OU=GETVPN
 revocation-check none
 rsakeypair pkiGM1
!
crypto pki certificate chain GETVPN
 certificate 6E
 certificate ca 01
!
crypto isakmp policy 10
 encr aes
 group 2
 authentication rsa-sig
!
crypto gdoi group getvpn
 identity number 1234
 server address ipv4 100.1.1.1
 server address ipv4 100.1.1.5
!
crypto map getvpn-map 10 gdoi
 set group getvpn
!
interface FastEthernet0/0
 description LAN interface
 ip address 10.1.11.1 255.255.255.0
 ip tcp adjust-mss 1360
!
interface FastEthernet0/1
 description WAN interface to PE
 ip address 100.1.1.9 255.255.255.252
```



```
 ip flow ingress
 load-interval 30
 duplex auto
 speed auto
 crypto map getvpn-map
!
router bgp 1111
 no synchronization
 bgp log-neighbor-changes
 network 10.1.11.0 mask 255.255.255.0
 network 100.1.1.8 mask 255.255.255.252
 neighbor 100.1.1.10 remote-as 1000
 no auto-summary
```

A.3 IOS Certificate Authority

A.3.1 Configuration

```
!
crypto pki server GETVPN
 database level names
 issuer-name CN = GET, OU = NSITE, O = CISCO, L = RTP, ST = NC
 grant auto
 lifetime crl 4
 lifetime certificate 730
 lifetime ca-certificate 1825
 database url flash:
!
crypto pki trustpoint GETVPN
 revocation-check crl
 rsakeypair GETVPN
!
crypto pki certificate chain GETVPN
 certificate ca 01
```

A.3.2 Verification

The CA configuration can be verified using the following command

```
DGVPN-CA# sh crypto pki server
Certificate Server GETVPN:
    Status: enabled
    State: enabled
    Server's configuration is locked  (enter "shut" to unlock it)
    Issuer name: CN = GET, OU = NSITE, O = CISCO, L = RTP, ST = NC
    CA cert fingerprint: FFD61C4E F12676BA FAADEFD4 E205EA6B
    Granting mode is: auto
    Last certificate issued serial number: 0x70
    CA certificate expiration timer: 18:11:56 EDT Jun 10 2012
    CRL NextUpdate timer: 14:19:08 est Feb 4 2008
    Current primary storage dir: flash:
    Database Level: Names - subject name data written as <serialnum>.cnm
```


Appendix B Abbreviations and Acronyms

The following table lists some common abbreviations and acronyms used to discuss GET VPN. Some common abbreviations and acronyms are not expanded in the text, but are included here for reference. Such terms are marked with an asterisk.

Abbreviation or Acronym	Expansion
3DES	Triple Data Encryption Standard (DES)
AAA	authentication, authorization, and accounting
ACL	access control list
AES	Advanced Encryption Standard
CA	certificate authority
CE	customer edge (device)
CLI*	command line interface
CM	Central Manager
COOP	Cooperative (Protocol)
CoS	class of service
CPE	customer premises equipment
DC	data center
DHCP*	Dynamic Host Configuration Protocol
DMVPN	Dynamic Multipoint VPN
DPD	dead peer detection
DSCP	Differentiated Services Code Point
DSL	digital subscriber line
DSLAM	digital subcriber line access multiplexer
FIB	Forwarding Information Base
FWSM	Firewall Switching Module (6500/7600)
GDOI	Group Domain Of Interpretation
GET VPN	Group Encrypted Transport VPN
GM	group member
GRE	Generic Routing Encapsulation
GUI*	graphical user interface
HA	high availability
HIPAA	Health Insurance Portability and Accountability Act
HSRP	Hot Standby Router Protocol
ICMP	Internet Control Messaging Protocol
IKE	Internet Key Exchange

Abbreviation or Acronym	Expansion
IP*	Internet Protocol
IPsec	IP security
ISP*	Internet service provider
KEK	key encryption key
KS	key server
L2TP	Layer 2 Tunneling Protocol
MAC*	Media Access Control
MPLS	Multiprotocol Label Switching
MSDP	Multicast Source Discovery Protocol
MTU	maximum transmission unit
NAT	network address translation
NTP	Network Time Protocol
OSPF*	Open Shortest Path First
PE	provider edge
PfR	performance routing
PKI	public key infrastructure
PSK	pre-shared keys
QoS	quality of service
SA	security association (IPsec)
SADB	security association database
SLB	server load balancing
SNMP*	Simple Network Management Protocol
SP	service provider
TBAR	time-based anti-replay
TEK	traffic encryption key
TFTP	Trivial File Transfer Protocol
UDP	User Datagram Protocol
VIP	virtual IP address
VLAN*	virtual local area network
VoIP*	voice over IP
VPN	virtual private network
VRF	virtual routing and forwarding
VSA	VPN Services Adapter (7200)
VTI	Virtual Tunnel Interface
WAAS	Wide Area Application Services