Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide

Haseeb Niazi, Nipul Shah, Biao Zhou, Varun Sethi

Shashi Shastry, Rudresh Veerappaji

August 2018

Contents

1. About Group Encrypted Transport Virtual Private Networks 3

1.1 Key GETVPN Benefits 4
1.2 Technology Overview 4
1.3 GETVPN Solution Positioning 9
1.4 GETVPN Solution Comparison 10
1.5 Further Reading 11

2. GETVPN Configuration 12

2.1 Implementing GETVPN 12
2.2 KS Configuration 13
2.3 GM Configuration 22
2.4 COOP KS Configuration 23
2.5 G-IKEv2 Configuration 27
2.6 Suite-B Support for GETVPN 30

3. GETVPNSystem Design 31

3.1 Platform Support 31
3.2 IOS Software Releases 31
3.3 KS Selection 32
3.4 GM Selection 32
3.5 KS Design Considerations 33
3.6 GM Design Considerations 46
3.7 COOP Design Considerations 56
3.8 Designing Around MTU Issues 89
3.9 VRF-Aware GETVPN 90
3.10 GETVPN Support for IPv6 in the Data Plane 100
3.11 GETVPN Support for LISP 103
3.12 GETVPN GDOI Bypass 105
3.13 GETVPN Routing Awareness 105
3.14 IPsec Inline Tagging for Cisco TrustSec on GETVPN 106
3.15 GETVPN Software versioning 108
3.16 GM Removal and Policy Replacement 111

4. Enterprise Deployment 116

4.1 DC and Branch Designs 116
4.2 DC Design 118
4.3 Branch Design 131
4.4 Deploying GETVPN 159

5. Provisioning, Verification, and Monitoring 165

5.1 Deploying GETVPN using CSM 165
5.2 GETVPN Syslog Capabilities 173
5.3 GDOI Event Trace 176
5.4 GETVPN Verification 178
5.5 SNMP Monitoring using GDOI MIB 191

Appendix A. Complete Configurations for Section 2 196

A.1 Using Pre-Shared Keys 199
A.2 Using Public Key Infrastructure (PKI) 203
A.3 IOS Certificate Authority 207
A.4 G-IKEv2 Configuration Using PKI 209

Appendix B. Steps to upgrade Key Servers and Group Members 214

Appendix C. Steps to change RSA Keys on Key Servers 215

Appendix D. Recent Features and Enhancements 217

Appendix E. Abbreviations and Acronyms 218

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 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 (GETVPN) 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 GETVPN networks, there is no need to negotiate point-to-point IPsec tunnels between the members of a group, because GETVPN is "tunnel-less."

The IETF standard RFC-6407 Group Domain of Interpretation (GDOI) is an integral part of GETVPN. The GDOI protocol was first introduced in 12.4(2)T but the GETVPN solution has had several enhancements over the newer releases. See 1.2.1, "GDOI," for more information about GDOI. The G-IKEv2 protocol was introduced In Cisco IOS Release 15.5(1)T and Cisco IOS Release 15.5(1)S. See 1.2.9 "GETVPN G-IKEv2," for more information about G-IKEv2.

1.1 Key GETVPN Benefits

GETVPN 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, GETVPN integrates very well with features such as QoS and traffic engineering.

1.2 Technology Overview

The GETVPN 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, GETVPN solution relies on following core building blocks to provide the required functionality:

- GDOI (RFC 6407)
- Key servers (KSs)
- Cooperative (COOP) KSs
- Group Members (GMs)
- IP tunnel header preservation
- Group security association
- Rekey mechanism
- Time-based anti-replay (TBAR)
- G-IKEv2
- IP-D3P

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 GETVPN 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.”

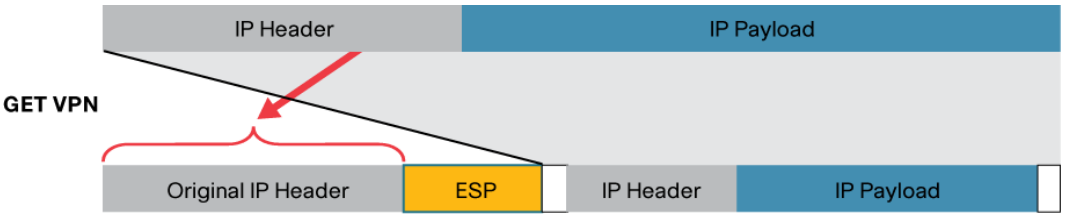
The GDOI protocol is protected by Internet Key Exchange (IKE) SA. All participating VPN gateways must authenticate themselves to the device providing keys using IKE. All IKE authentication methods, for example, pre-shared keys (PSKs) and public key infrastructure (PKI) are supported for initial authentication. 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. For more information about GDOI, refer to RFC 6407.

GDOI introduces two different encryption keys. One key secures the GETVPN 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 GETVPN, 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. 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 GETVPN 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.

Note: It is worth noting that tunnel header preservation seems very similar to IPsec transport mode. However, the underlying IPsec mode of operation with GETVPN 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 when used together with Tunnel Header Preservation and must not be used in GETVPN deployments where encrypted or clear packets might require fragmentation.

Note: Because of tunnel header preservation, GETVPN solution is very well suited for MPLS, Layer-2 (L2), or an IP infrastructure with end to end IP connectivity. However, GETVPN 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 Key Servers (KSs)

A key server (KS) is a device responsible for creating and maintaining the GETVPN 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 (pre-shared keys or PKI) and download the encryption policies and keys required for GETVPN 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, irrespective of 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 opposed 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 cannot be configured as a GM

1.2.4 Group Members (GMs)

A GM is a device responsible for actual encryption and decryption i.e. a device responsible to handle GETVPN data plane. A GM is only configured with IKE 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 GETVPN 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, GETVPN uses the concept of group SA. All members in the GETVPN 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 GETVPN GMs.

Note: In a GETVPN group, up to 100 ACL entries (permit and deny) can be used to define interesting traffic for encryption. If an ISM-VPN card is used on Cisco ISR G2, the recommended KS ACL size is 70 entries, whereas it is 32 entries for local exception ACL on GM.

It is a best practice to summarize interesting traffic to as few permit entries as possible, and to build symmetric policies. Unlike traditional IPSec policy, where source and destination address ranges must be uniquely defined, GETVPN is optimized when the source and destination address range are the same. This minimizes the number of policy permutations, making GETVPN 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. GETVPN 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 when only 5% of the original TEK lifetime remains. Registration must be completed 30 seconds before the existing IPsec SAs expire as GM's that have successfully received rekey or re-registered will start using the new IPSec SA when 30 seconds remain in the previous TEK's lifetime. The 30 second window provides a graceful roll-over period for traffic encrypted with the previous TEK to clear the network before the deletion of the expiring TEK. 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. The process repeats through each of the ordered set of KS until all KS have been attempted. The GM will restart the registration process to the first KS in the ordered set until registration is successful.

Note: Volume-based rekey is disabled for SA setup using GETVPN crypto maps.

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 once to GM's that successfully receive the rekey and the KS 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 scheduled rekeys, the KS removes the GM from its active GM database and stops sending rekey messages to that GM.

Note: Unicast rekey retransmission occurs 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 GETVPN control plane.

1.2.7 COOP KSs

The KS is the most important entity in the GETVPN 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 keys and distributing configured 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 possible 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) on a 20 second interval. If a secondary KS does not hear from the primary KS over a 30 second interval, the secondary KS tries to contact the primary KS and requests updated information. If the secondary KS does not hear from the primary KS over a 60 second interval, a COOP reelection process 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: Periodic DPD must be enabled on the ISAKMP SAs between the KSs if COOP is configured. This way the primary KS can track and display the state of the other secondary KSs. Periodic DPD between GM and KS is not required.

1.2.8 Time Based Anti-Replay (TBAR)

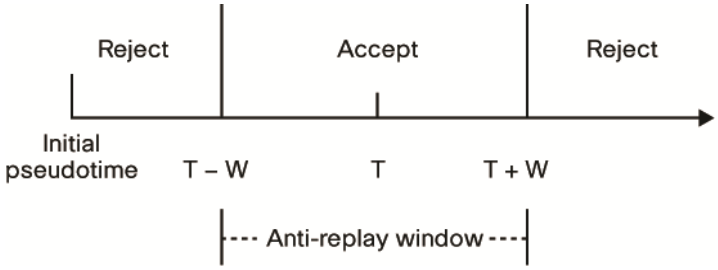
In traditional IPsec solutions, anti-replay capabilities prevent a malicious third party from capturing IPsec packets and relaying those packets at a later time to launch a denial of service attack against the IPsec endpoints. These traditional IPsec solutions use a counter-based sliding window protocol: 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 GETVPN, 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 GETVPN 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, which by default is sent every 7200 seconds or 2 hours. 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 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 enabled group wide on the KS.

1.2.9 GETVPN G-IKEv2

Internet Key Exchange Version 2 (IKEv2) provides improvisations to ISAKMP infrastructure with lessons learned from the industry. IKEv2 reduces network latency, reduces complexity in message exchanges, improves interoperability and reliability, and fixes cryptographic issue in HASH authentication. GETVPN combines IKEv2 protocol with IPsec to provide an efficient method to secure IP multicast traffic or unicast traffic through the GETVPN G-IKEv2 feature. This feature provides a complete IKEv2 solution across all of Cisco's VPN technologies.

The G-IKEv2 protocol provides a mechanism for a group member (GM) to download policy and keys from a key server (KS). These policy and keys are used to secure communication among GMs in a group. G-IKEv2 is a new model to secure group communication between remote locations in an enterprise private WAN.
GETVPNGETVPN

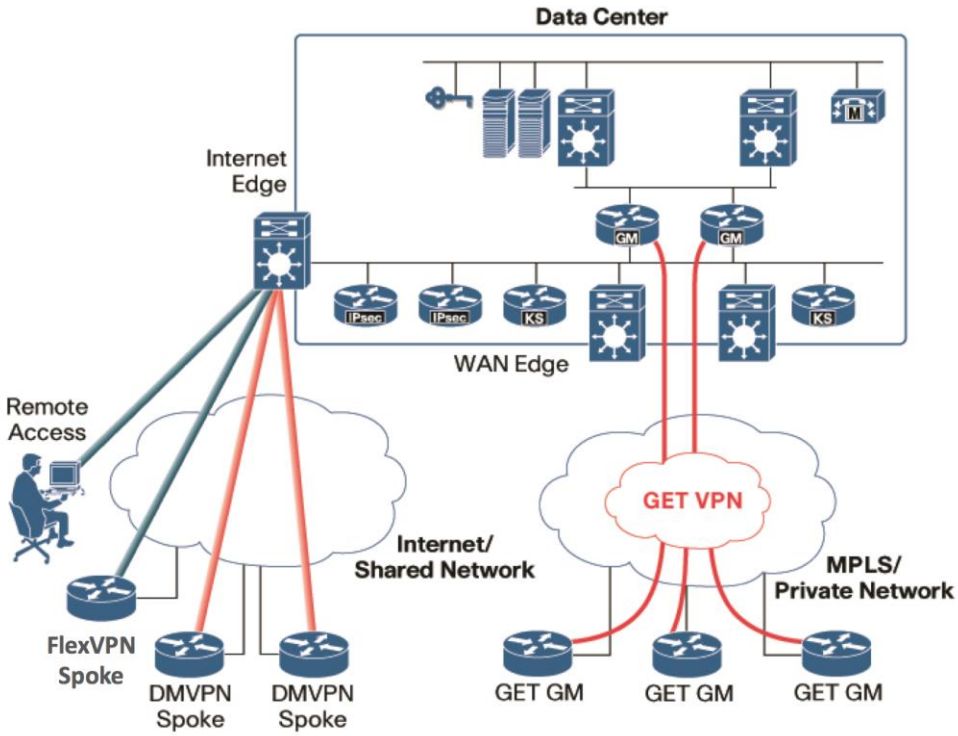
1.2.10 GETVPN Interoperability—IP-D3P

The IP-D3P feature implements IP Delivery Delay Detection Protocol on Cisco software. IP-D3P, enabled on the crypto data plane, uses the system clock of the GMs to create or verify the IP-D3P datagram's timestamp. In many cases, the system clock is set from an external protocol, for example, Network Time Protocol, to maximize the likelihood that the system clocks of both sender and receiver are synchronized.
IP datagrams are subject to delivery delay attack, in which a host or gateway receives datagrams that are not fresh. IP-D3P datagram consists of a header and an IP payload. The IP-D3P header includes a timestamp, which receivers of the packet use to determine whether or not the packet has been recently generated. Receivers compare the timestamp delivered in the IP packet to their local time and decide if the IP packet must be accepted.
Configurations and verification with examples for the GETVPN Interoperability—IP-D3P feature are described in GETVPN Configuration Guide.

1.3 GETVPN Solution Positioning

Figure 3 illustrates the positioning of the GETVPN solution.

Figure 3. GETVPN Solution Positioning



As noted, GETVPN is well suited for deployments over private WAN networks, allowing the utilization of underlying MPLS/shared IP networks using header preservation. GETVPN is not well suited for deployment over the Internet, so a tunnel-based IPsec solution such as DMVPN, FlexVPN, or Site-to-Site VPNs should be deployed over the Internet.

1.4 GETVPN Solution Comparison

Table 1 provides a basic comparison of DMVPN, FlexVPN and GETVPN technologies. Consult the detailed documentation about these technologies for further information.

Table 1. GETVPN Solution Comparison

Solution	DMVPN	FlexVPN	GETVPN
Infrastructure Network	Public Internet Transport	Public Internet Transport	Private IP Transport
Network Style	Hub-Spoke and Spoke-to-Spoke; (Site-to-Site)	Hub-Spoke and Spoke-to- Spoke; (Client-to-Site and Site-to-Site)	Any-to-Any; (Site-to-Site)
Routing	Dynamic routing on tunnels	Dynamic routing on tunnels or IKEv2 routing or IKEv2 Dynamic routing	Dynamic routing on IP WAN
Failover Redundancy	Route Distribution Model	Route Distribution Model	Route Distribution Model
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 GETVPN technology and discusses key aspects of the technology, it should not be considered an in-depth technology primer. For further details on the GETVPN technology, refer to following documentation:

GETVPN Documentation on

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

GETVPN Feature Documentation

IOS platforms: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_getvpn/configuration/15-mt/sec-get-vpn-15-mt-book.html

IOS-XE platforms: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_getvpn/configuration/xe-3s/sec-get-vpn-xe-3s-book.html

2. GETVPN Configuration

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

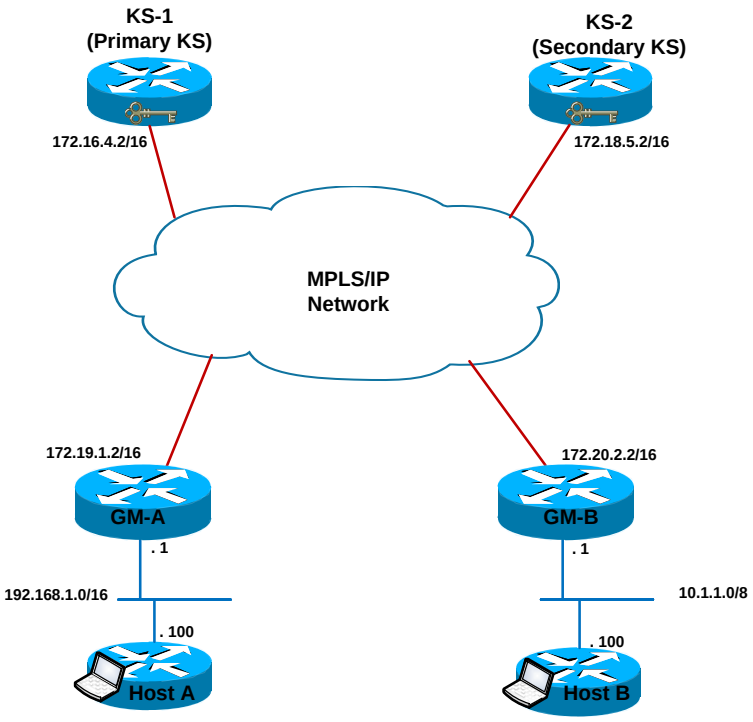
- Key servers (KSs) and group members (GMs) for GDOI and G-IKEv2
- Pre-shared key (PSK) and public key infrastructure (PKI)
- Unicast and multicast rekey
- Cooperative (COOP) KSs
- IPsec Inline Tagging for Cisco TrustSec
- GETVPN GDOI Bypass
- GETVPN CRL Checking
- GETVPN Routing Awareness

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

2.1 Implementing GETVPN

Figure 4 illustrates a high-level GETVPN system configuration topology.

Figure 4. System Configuration Topology



The topology in Figure 2 is used to setup the GETVPN network. The IP VPN core interconnects VPN sites as shown in the figure. The CE/CPE routers (GMs A and B) 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

Internet Key Exchange (IKE) 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 128
  hash sha256
  group 14
```

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 172.18.5.2
  crypto isakmp key Cisco address 172.19.1.2
  crypto isakmp key Cisco address 172.20.2.2
```

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

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 IDCertKey modulus 2048
The name for the keys will be: IDCertKey


% The key modulus size is 2048 bits
% Generating 2048 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://172.17.1.2:80
  subject-name OU=GETVPN
    revocation-check none
    auto-enroll 70
    rsakeypair IDCertKey 2048
```

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 pki certificates
Certificate
Status: Available
Certificate Serial Number: 0x6E
Certificate Usage: General Purpose
```

```
Issuer:
cn=GET
ou=ENG
o=CISCO
l=RTP s
t=NC
Subject:
Name: KS
hostname=KS
ou=GETVPN
Validity Date:
start date: 10:15:09 EST Apr 25 2012
end date: 10:15:09 EST Apr 25 2017
Associated Trustpoints: GETVPN
CA Certificate
Status: Available
Certificate Serial Number: 0x1
Certificate Usage: Signature
Issuer:
cn=GET
ou=ENG
o=CISCO
l=RTP
st=NC
Subject:
cn=GET
ou=ENG
o=CISCO
l=RTP
st=NC
Validity Date:
start date: 17:11:56 EST Jun 12 2012
end date: 17:11:56 EST Jun 10 2017
```

```
Associated Trustpoints: GETVPN
```

Note: It is recommended to configure **auto-enroll** so that routers can re-enroll with the CA server before the expiry of the certificate. Without the auto-enroll configuration, the network administrator will have to manually re-enroll all routers before the certificates expire. If the KS refuses to accept expired certificates during registration, then the GMs will suffer an outage, causing a disruption in the network

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 GETVPN 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.

```
KS(config)#crypto key generate rsa general-keys label getvpn-export-general
modulus 2048 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 172.16.4.2
```

The GETVPN 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 GETVPN (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 GETVPNGETVPN. 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.

GETVPN supports both Symmetric and Asymmetric policies on the key server.

For example, with symmetric policies the following can be configured on the key server:

Symmetric:

permit ip any any

Symmetric:

permit ip 10.0.0.0/8 10.0.0.0/8

However for Asymmetric permit policies to work correctly, a reciprocal policy has to be configured on the key server as follows:

Asymmetric:

```
permit ip 10.0.0.0/8 any
permit ip any 10.0.0.0/8<==== reciprocal required for permit statements
```

Asymmetric:

```
permit ip 10.0.0.0/8 172. 16.0.0/16
permit ip 172. 16.0.0/16 10.0.0.0/8<==== reciprocal required for permit
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
```

2.2.4.1 Changes in the behavior of the GETVPN crypto policy

This section discusses the differences in the behavior of the access-list policy installed on the GM under the following 2 conditions:

1. **When the access-list is downloaded from KS**

2. When the access-list is configured locally on the GM

1. When access-list is downloaded from KS:

Downloaded access-list policy will be installed on the GM in a centralized model. In this model, both the inbound and outbound IPsec SAs will be installed with the same source-destination addresses.

For example:

Consider the following topology and ACL rule configured on the KS

Host/Network A----------GM1==========GM2----------Host/Network B

```
access-list 101 permit ip host A host B
```

The ACL will install the following policy on GM1 and GM2:

Outbound policy: Encrypt traffic from host A to host B

Inbound policy: Decrypt traffic from host A to host B

Based on the above policy, traffic from A to B will be encrypted on GM1 and decrypted on GM2. However, traffic from B to A will go in clear. In order to secure traffic from B to A, a reciprocal ACL rule needs to be configured on the KS:

```
access-list 101 permit ip host B host A
```

The reciprocal ACL will install the following policy on GM1 and GM2:

Outbound policy: Encrypt traffic from host B to host A

Inbound policy: Decrypt traffic from host B to host A

2. Behavior when access-list policy is locally configured on the GM:

Locally configured access-list policy will be installed on the GM in a site-to-site model. In this model, the outbound IPsec SA will be installed with the same source-destination address as the ACL rule, while the inbound IPsec SA will be installed with the inverted source-destination addresses. Please note that only deny access-list entries can be configured locally on GMs.

For example:

Consider the following topology and ACL rule configured on the locally GM1

Host/Network C----------GM1==========GM2----------Host/Network D

```
access-list 101 deny ip host C host D
```

The ACL will install the following policy on GM1:

Outbound policy: Do NOT encrypt traffic from host C to host D

Inbound policy: Do NOT decrypt traffic from host D to host C

Based on the above policy, traffic from C to D will be sent in clear, while traffic from D to C will bypass decryption.

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

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 128
lifetime 1200
authentication pre-share
group 14
crypto isakmp key Cisco address 172.16.4.2
crypto isakmp key Cisco address 172.18.5.2
```

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 GETVPN

Applying crypto map to the WAN interface enables GDOI.

```
interface Ethernet0/0
 description WAN interface to MPLS PE
```

```
    ip address 172.19.1.2 255.255.0.0
!
crypto map enabled on WAN interface
```

Note: crypto map getvpn-mapApart from Ethernet interfaces, GDOI/GKM crypto maps are supported on tunnel interfaces, LISP interfaces, Virtual-PPP interfaces (except ASR1000 – CSCvh50336) and port-channel interfaces on IOS-XE platforms.

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 GETVPN. 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 2048 exportable
% The key modulus size is 2048 bits
% Generating 2048 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 GETVPN 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 172.18.5.2
```

2.4.2.2 Secondary KS

```
!
Crypto isakmp keepalive 15 periodic
```

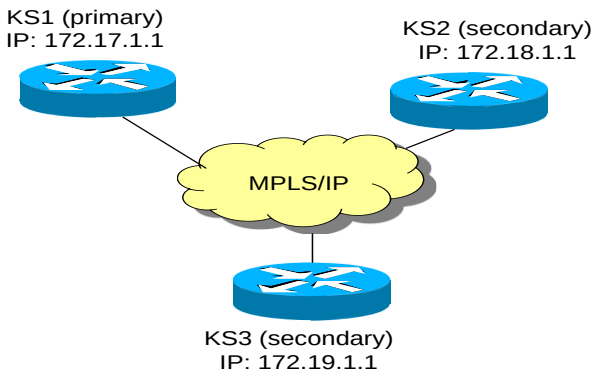
```
!
crypto gdoi group getvpn
server local

! 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 172.16.4.2
```

2.4.2 Adding a New Cooperative Key Server

The following paragraph describes the method to follow in order to add a new cooperative key server into an existing key server cooperative group.

To minimize disruptions to the existing cooperative key server network, a new cooperative key server should not be included until its ip address is known to all the existing key servers. The introduction is carried out by adding the new ip address to the peer list of the existing key servers. It is recommended that the process of the configuration changes should start with Secondary key servers first before the final change is made in the Primary.

For example: Consider a cooperative KS network with KS1 as primary and KS2 as secondary KS. Follow these steps to add a new KS (KS3) to the cooperative KS network:



- Verify the reachability of KS3 from KS1 and KS2.
- Add the ip address of KS3 to KS2.
- Add the ip address of KS3 to KS1.

2.5 G-IKEv2 Configuration

It is assumed that all connectivity (default routes, routing protocols (RPs) are already set up.
All GDOI constructs discussed in the previous sections apply to G-IKEv2 as well. In the subsequent sections the specific configurations needed to enable G-IKEv2 on GM and KS are described. Note that GKM is the generalized name for the technology to cover both GDOI and G-IKEv2 constructs.

Figure 5. GETVPN Architecture through G-IKEv2 Protocol

As shown in the figure, there are three steps for GM's to download group security policy and establish secure connectivity between the other GMs in the group.

1. Initial GM Registration Request—A GM initiates a registration request to the KS, sending preferred cryptographic algorithms (in SAi payload), Diffie-Hellman public number (in initiator's KE phase 1 payload) and nonce (random number used to guarantee liveness in Initiator's Nonce payload). This message is the same as the IKE_SA_INIT request message defined in the IKEv2 RFC.

2. Group Secured Communication—The GM's do not directly establish pair-wise IPsec tunnels with one another, but use the group IPsec policy and keys to secure communication between all GM's in a particular group.
3. KS Rekey—The KS distributes new group keys to group members as needed using the G-IKEv2 group maintenance channel via unicast or multicast communication. Rekey is optional in G-IKEv2.

G-IKEv2 allows GETVPN to do the following:

1. Inherit the implicit advantages of IKEv2 & the IOS IKEv2 implementation
2. Keep the underlying technology up-to-date for future development
3. Provide a complete IKEv2 package across Cisco's VPN security offerings

2.5.1 GDOI to GKM Configuration Syntax Changes

Group Key Management or GKM is a generalized name or acronym for the technology going forward. G-IKEv2 does not include the Domain of Interpretation, therefore, a generic abbreviation **gkm** referring to Group Key Management is used for a group that can use either GDOI or G-IKEv2 protocols for registration and rekey. The gkm configuration syntax was introduced in Cisco IOS Release 15.5(1)S and Cisco IOS Release 15.5(1)T, wherein both commands **crypto gdoi** and **crypto gkm** were available. However, the GDOI keyword has been deprecated as of Cisco IOS Release 15.5(3)S or Cisco IOS Release 15.5(3)M and replaced by the **gkm** keyword which supports both GDOI & G-IKEv2.

The GKM version in Cisco IOS Release 15.5(1)S and Cisco IOS Release 15.5(1)T is as follows:

- Key Server: 1.0.13
- Group Member: 1.0.12

The difference in KS & GM versions is due to the GDOI Rekey ACK & IP-D3P features only existing on the KS in the first release.

2.5.2 G-IKEv2 Supported Features and Restrictions

Refer to GETVPN Configuration Guide for the list of G-IKEv2 supported features and Restrictions.

2.5.2.1 IKEv2 Profile Configuration

Both KS & GM must specify an IKEv2 profile. The IKEv2 profile is used for initiating group member and processing key server registrations & reregistrations. Either Pre Shared Key (PSK) or Public Key Infrastructure (PKI) can be used. The IKEv2 Smart Defaults works seamlessly with G-IKEv2.

To configure IKEv2 profie,

```
crypto ikev2 profile gkm-gikev2
  match address local 10.0.0.1
  match identity remote address 10.0.0.5
  authentication local pre-share key
  authentication remote pre-share key
  keyring local kyr_aaa

crypto ipsec profile gkm-ipsec-profile
  set transfor-set TEK
  set ikev2-profile gkm-gikev2
```

2.5.2.2 G-IKEv2 Key Server Configuration

GDOI & G-IKEv2 are enabled or disabled independently in each group. This configuration does the following:

- Specifies which type(s) of registration can be serviced by the KS
- Specifies which type(s) of rekey will be sent by the KS

Note: The default is GDOI-enabled and G-IKEv2 disabled.

```
crypto gkm group gkm-grp1
 server local
  gdoi
  gikev2 gkm-gikev2
```

2.5.2.3 G-IKEv2 Group Member Configuration

Only one protocol, either GDOI or G-IKEv2, can be specified in each group on a GM.

```
crypto gkm group gkm-grp2
  client protocol gikev2 gkm-gikev2
```

This configuration does the following:

- Specifies which registration protocol will be used by the GM
- Specifies which rekey protocol will be expected by the GM

Refer Appendix A.4 for complete G-IKEv2 configurations.

Note: The default client protocol is gdoi and not g-ikev2.Refer to the appendix () for detailed configuration example.

2.5.3 GDOI to G-IKEv2 Migration

Over a period of time, you may want to upgrade and migrate your key servers and group members to G-IKEv2 which is recommended. Migration from GDOI to G-IKEv2 for an entire GETVPN group requires careful planning. You cannot migrate all your group members at the same time. The migration entails allowing GDOI group members and G-IKEv2 group members to communicate using the same traffic

encryption key (TEK) while using different control plane protocols—GDOI and G-IKEv2. A GDOI to G-IKEv2 migration sequence includes the following:

- Backward compatibility—The new Cisco IOS software image containing the GETVPN G-IKEv2 feature must support existing GDOI features and must be consistent with for earlier releases of GDOI features for Cisco IOS software.
- Service upgrade—The recommended sequence for changing the Cisco IOS software image is secondary key server, primary key server, and group member.
- Service downgrade—The recommended sequence for changing the Cisco IOS software image is group member, secondary key server, and primary key server.

Detailed migration procedures with examples are illustrated in the G-IKEv2 configuration guide.

2.6 Suite-B Support for GETVPN

The National Security Agency (NSA) along with the National Institute of Standards and Technology (NIST) has proposed a set of cryptographic algorithms named Suite-B as a standardized & interoperable base for cryptographic operations. These cryptographic algorithms include:

- AES-GCM or AES-GMAC with 128 and 256-bit keys,
- Elliptic-Curve Digital Signature Algorithm (ECDSA) for digital signatures,
- Elliptic-Curve Diffie-Hellman (ECDH) for key agreement, and
- Secure Hash Algorithm 2 (SHA-256 and SHA-384) for message digest / integrity.

The GETVPN Support with Suite B feature allows these cryptographic algorithms to be used with GDOI and GETVPN in various ways, including the use of SHA-2/HMAC-SHA-2 and AEC-GCM/AES-GMAC.

Refer to GETVPN Configuration Guide for details on GETVPN Suite-B implementation, configuration and verification with examples.

3. GETVPN System Design

To design a GETVPN 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 GETVPN 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 GETVPN network is another important design consideration. This chapter addresses these design considerations in detail.

3.1 Platform Support

Table 2 lists the most common platforms in the Cisco enterprise portfolio having encryption capability and modules targeting GETVPN support.

Table 2. Currently Supported GETVPN Platforms

Product Line	GM	KS
841M	Yes	Not recommended
87x	Yes	Not recommended
880/890	Yes	Not recommended
ISRv	Yes	Not recommended
1100	Yes	Not recommended
1900[1]	Yes	Yes (for 1941)
2900[2]	Yes	Yes
3900/E[2]	Yes	Yes
ISR 4200	Yes	Not recommended
ISR 4300	Yes	Yes (only 4351)
ISR 4400	Yes	Yes
ASR1000	Yes	Yes
ASR1000-X	Yes	Yes
ASR1000-HX	Yes	Yes
CSR1000v	Yes	Yes

3.2 IOS Software Releases

Please contact your Cisco representative for the most updated IOS release recommendations.

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 2 shows the supported platforms for Key Server, refer to GETVPN CCO documents on cisco.com for the KS scalability numbers. Please contact your Cisco representative for more detailed KS performance and scalability data.

The 8000 GM Scale Improvement feature supports optimization of the Cooperative Protocol (COOP) announcement messages by increasing the number of Group Members (GM) to 8000 from the earlier scale of 4000. This feature is available from Cisco IOS Release 15.5(1)T and Cisco IOS 15.5(1)S. To use this feature to increase the GM scale from 4000 to 8000, use the protocol version optimize command under the gdoi/gkm group on the Key Server.

Refer to the 8K GM Scale Improvement document for more details on the procedure.

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

Please contact your Cisco representative for a detailed GM performance data.

3.4.2 Number of Security Associations

Some platforms (e.g. 800 series) support only a limited number of security associations (SAs).In GETVPN, 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 GETVPN policy results in creation of 100 SAs on the GM (50 permit entries in the encryption ACL creates 50 outbound SA and 50 inbound SAs), the platform should be able to handle 300 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 Typical choice of platforms in an Enterprise deployment

Figure 6 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 4000 and ASR1000 models at branches depends on the connection type and throughput requirements, among other parameters.

Figure 6. 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
 hash sha256
 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 172.16.4.2
```

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

Before using Suite B for traffic encryption key (TEK), ensure that all GM platforms used in the deployment support Suite B. For more information regarding platform platform support for Suite B, see: https://www.cisco.com/c/en/us/support/docs/security-vpn/ipsec-negotiation-ike-protocols/116055-technote-ios-crypto.html

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 the recommended 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; however, setting the TEK lifetime too low leads to excessive key generation and key overlap. Setting the TEK lifetime to 900 seconds 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)]
The example provided leads to a TEK lifetime overlap of 940 seconds. The new TEK is pre-positioned 940 seconds prior to the expiration of the previous TEK. While the new TEK is distributed prior to its use, the GM's will always encrypt with the TEK with the least remaining TEK lifetime. The GM's will start encrypting with the new TEK when 30 seconds reaming in the previous TEK's lifetime.

3.5.4 Long SA

The GETVPN Resiliency features enhances troubleshooting capability and operation stability in GETVPN networks so that traffic disruption is prevented or minimized when errors occur. One of the resiliency feature is Long SA Lifetime.
By choosing a longer TEK and KEK lifetime, the KS rekey frequency is reduced, thus, improving GETVPN operation stability by:

- Lowering the chance of rekey failures
- Lowering the chance of rekey happening during COOP-KS split/merge
- Lowering the KS CPU usage due to rekey

To configure Long SA Lifetime for TEK:

```
enable
 configure terminal
  crypto ipsec profile gdoi-p
  set security-association lifetime days 15
```

To configure Long SA Lifetime for KEK:

```
enable
 configure terminal
  crypto gdoi group GET
   identity number 3333
   server local
   rekey lifetime days 20
```

Refer to the Long SA document for more information.

3.5.5 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. A recommended best practice is to insure that any bootstrap management and control protocols are denied encryption on the local GM during early deployment phases. A global deny policy may be constructed for all GM's and deployed from the KS to insure that all management and control bootstrap protocols are never encrypted.

Note: ISR GMs do not need explicit deny ACL entries for IKE/GDOI traffic. The deny entries are required for ASR GMs running IOS-XE releases prior to XE3.8 or 15.3(1)S.

Note: Cisco ASR 1000 GMs do not support access-lists that have discontiguous wildcard netmasks. For example, the following access-list entry is not supported:

access-list 101 permit ip 10.0.0.1 0.255.255.0 192.168.0.0 0.0.255.255
Please design the network such that the TEK ACL will have contiguous netmasks. For example, the above access-list entry would be modified as:
access-list 101 permit ip 10.0.0.1 0.255.255.255 192.168.0.0 0.0.255.255

3.5.6 CRL Checking

When PKI is used for authentication, certificates are validated during the initial session establishment between GM and KS. If a certificate gets revoked after the session is established, these existing sessions are not affected by the revocation of that certificate. The GETVPN CRL Checking feature notifies KSs through public key infrastructure (PKI) when a new CRL is available for a configured trustpoint. The KS creates a new Key Encryption Key (KEK) and sends a reauthentication message to the group member devices, which print a syslog message, delete the current KEKs, and reregister to the KS.

The GETVPN CRL Checking feature provides the following

- New reauthentication configuration for GETVPN to specify a PKI trustpoint.
- PKI notifies GETVPN when a new CRL is available for download at the PKI server.
- GETVPN verify if new CRL matches the configured PKI trustpoint.
- If the CRL matches, GETVPN KS start a reauthentication process to have all GMs reregister.
- The reauthentication process will be similar to a GM removal.

You need to configure several components prior to enabling the GETVPN CRL Checking feature. These include:

- A defined PKI certificate authority (CA) so that group members and key servers are PKI clients and, therefore must enroll to get certificates.
- KSs configured to have certificate revocation list (CRL) checking enabled in PKI.
- KSs configured to download the CRL when it is available on the CA and on a first-needed basis. This means that the KSs download the CRL following the first group member (GM) registration after the new CRL is available.
- CRL checking disabled on the group member devices for PKI.
- Internet Key Exchange (IKE) authentication set to certificates.

In the following example, KSs are configured to download the CRL when the first group member registration occurs after a new CRL is available on the trustpoint CA named mycert:

```
ip domain name cisco.com
ip http server
crypto pki trustpoint mycert
 enrollment url http://10.1.3.1:80
 revocation-check crl

crypto identity abcd
 fqdn ut01-unix5.cisco.com
 fqdn ut01-unix6.cisco.com
```

```
crypto gkm group gdoi-group1
 server local
 authentication identity abcd
Crypto gdoi group gdoi_group1
 server local
  registration periodic crl trustpoint mycert
```

Refer to CRL Checking document for more information.

3.5.7 Key Encryption Key Lifetime

The key encryption key lifetime should be left at the default of 86400 seconds. Since the KEK is used to encrypt the control plane messages between the KS and GM. Changing the KEK value frequently subjects the GM to possible rekey misses and subsequently requiring the GM to re-register more frequently than is necessary. It is recommended that the KEK lifetime should always be at least double the TEK lifetime.

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.

3.5.8 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.
- 3 schdeduled rekeys

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
```

The periodic reminder sync-up rekey functionality in the key server (KS) lets you to send periodic reminder rekeys to group members (GMs) who do not respond with an acknowledgment (ACK) in the last scheduled

rekey. This functionality in combination with the long SA lifetime functionality is effective for a KS to synchronize with GMs when they miss a scheduled rekey before the keys rollover. In a KS group configuration, a new keyword periodic is added to the rekey retransmit command when configuring the rekey retransmission.

Each periodic rekey increments the sequence number, similar to rekey retransmissions. The GM is removed from the database on the KS after 3 scheduled rekeys (not retransmissions) for which the GM does not send an ACK.

If periodic is used, here is the configuration snippet:

```
crypto gdoi group group1
 identity number 3333
 server local
 rekey retransmit 10 periodic
```

To verify this value, use show crypto gdoi ks rekey.

3.5.9 IKEv2 profile based G-IKEv2 authorization

In a G-IKEv2-enabled network, a Key Server (KS) authorizes a Group Member (GM) on the GM's identity. The identities currently supported by KS are IP address, fully qualified domain name (FQDN), distinguished name (DN), email-id and key-id. The identity of a GM used for authorization is same as the identity used in IKEv2 protocol. This G-IKEv2 Enhancement for GETVPN feature leverages IKEv2 supported identities thereby enabling IKEv2 profile-based authorization on G-IKEv2 networks. The GM identity is used by IKEv2 and GKM on KS to authenticate the GM. IKEv2 uses its identity to bring up a session and GKM uses identity for GM registration. In case of GDOI-based networks, the authorization must be configured via the authorization command in the GDOI/GKM group.

Note: Although, GDOI and G-IKEv2 supports access control list configuration in **an** authorization list, IKEv2 does not support ACL in identity configuration. Therefore ACL is not supported on G-IKEv2 networks but is supported and applicable for GDOI networks.

To configure IKEv2 profile based G-IKEv2 authorization,

```
crypto ikev2 profile IKEv2-PROFILE
 match identity remote email sovm@cisco.com
 identity local email abc@gmail.com

<Similarly,>
 match identity remote fqdn cisco.com
 match identity remote address 10.10.10.1 255.255.255.255
 match identity remote key-id sovm
 identity local remote key-id sovm
 identity local remote fqdn cisco.com
```

Refer to the configuration guide for more information.

3.5.10 TBAR

TBAR should be configured on all platforms and counter-based anti-replay should **not** be configured.

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”.

3.5.11 IP-D3P IP-Delivery Delay Detection Protocol

The IP-D3P Support implements draft-weis-delay-detection-04.txt on GETVPN. IP-D3P uses the system clock of group members to create and verify the IP-D3P datagram’s timestamp. In most cases, the system clock is set from an external protocol, such as Network Time Protocol (NTP) – which is recommended, to synchronize the system clocks of the sender and receiver.

3.5.11.1 IP-D3P Support for Key Server

A new configuration command, **d3p**, in the GDOI local server configuration mode allows you to enable IP-D3P on a key server. The show crypto gkm ks command displays the IP-D3P parameters that are enabled on a key server.

3.5.11.2 IP-D3P Support for Group Member

Group members receive the IP-D3P parameters present in the rekey messages. Group members process the new GAP payload attributes—D3P-TYPE and D3P-WINDOWSIZE. The window-size, which must be used in IP-D3P for a group member, can be overwritten by using the client d3p command in the GDOI group configuration.

To ensure that all devices in a GETVPN network support GETVPN interoperability features, use the command “**show crypto gkm feature** *feature name*”. This displays the GDOI version running on each key server and group member in the network and information about whether the device supports GETVPN interoperability features, namely, D3P support on GETVPN Key Server and Internet-Draft ACK for Cisco GETVPN Key Server.

```
Device# show crypto gkm feature ip-d3p
Group Name: GETVPN1
    Key Server ID       Version   Feature Supported
```

```
        10.0.8.1         1.0.11         Yes
        10.0.9.1         1.0.10         No
    Group Member ID     Version    Feature Supported
        10.0.3.1         1.0.11         Yes
        10.65.9.2        1.0.10         No
```

3.5.11.3 Configuration

3.5.11.3.1 Enabling IP-D3P on a Key Server

To configure IP-D3P on KS,

```
crypto gkm group GETVPN
<..>
  sa d3p window msec 5000
```

3.5.11.3.2 Enabling IP-D3P on a Group Member

To configure IP-D3P on GM,

```
crypto gkm group GETVPN
<..>
  client d3p window sec 50
```

3.5.11.4 GDOI Rekey Acknowledgement

The Internet-Draft ACK for Cisco GETVPN Key Server feature implements the standards for rekey acknowledgment messages between non-Cisco group members and a key server, as defined in the GDOI GROUPKEY-PUSH Acknowledgment Message draft.

The GDOI GROUPKEY-PUSH Acknowledgment message, which is referred to as GDOI I-D Rekey ACK, differs from the Cisco unicast rekey acknowledgment message by defining an interoperable method for a group member to send a rekey acknowledgment to any key server in a group. The GDOI I-D Rekey ACK is not *encrypted* , as is the case with Cisco unicast rekey acknowledgment message, but is *integrity-checked* . When a key server sends a rekey message to group members for updating the keys and policies of a group, it is useful for a key server to know if all group members have received the rekey message and have successfully processed, installed, and responded to the new keys and policies.

To ensure that all devices in a GETVPN network support GETVPN interoperability features, use the command "**show crypto gkm feature** *feature name*". This displays the GDOI version running on each key server and group member in the network and information about whether the device supports GETVPN interoperability features, namely, D3P support on GETVPN Key Server and Internet-Draft ACK for Cisco GETVPN Key Server.

```
Device# show crypto gkm feature gdoi-interop-ack
Group Name: GETVPN2
    Key Server ID       Version   Feature Supported
        10.0.8.1         1.0.11         Yes
        10.0.9.1         1.0.10         No
    Group Member ID     Version   Feature Supported
        10.0.3.1         1.0.11         Yes
        10.65.9.2        1.0.10         No
```

To enable GDOI ID-Rekey ACK on a key server, configure as below.

```
crypto gkm group GET
<..>
```

`rekey acknowledgment interoperable<..>`For more details, refer to Interoperable Rekey Acknowledgement section on this configuration guide.

3.5.12 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, GETVPN also supports GM authorization as described in the following sections. GDOI authorization is based on ISAKMP identity sent by the GM. With PSK authentication, if GM sends ip address as identity only "authorization address" will be used to authorize the GM. With PKI, if GM sends DN or FQDN or hostname then "authorization identity" will be used to authorize the GM.

Using an IP address as an identity will bypass authorization matching a DN or hostname and vice versa. To ensure that only GMs with a specific DN can connect (and no GMs using another identity can connect), we must specify “deny any” in the authorization address ACL.

3.5.12.1 GM Authorization Using PSKs

GETVPN supports GM authorization using the **IP address in the IKE identity** when using PSKs. An ACL matching these IP addresses can be defined and applied to the GETVPNGETVPN group configuration on the KS. Any GM whose IP address (in IKE identity) matches the ACL, gets authorized successfully and can register to the KS. Else, KS rejects the GM registration request.

```
!
! On KS
!
crypto gdoi group getvpn
 server local
  authorization address ipv4 50
!
access-list 50 permit 10.1.1.10
access-list 50 permit 192.168.1.0 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: 10.1.1.9
```

3.5.12.2 GM Authorization using PKI

GETVPN 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 GETVPN 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 GETVPN 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.

3.5.13 Key Server Role Change

A new CLI command provides the user with a way to change the priorities of certain key servers and shift a key server from Secondary to Primary while demoting the old Primary to Secondary. Prior to the introduction of this command, “clear crypto gdoi” was the only way to change the key server roles. However, this is a more drastic command and causes existing policies to be deleted and also causes the key servers to go into the election mode.

WARNING: The command “clear crypto gdoi” when executed on the key server, clears all the existing registrations It will not force GM’s to re-register. KS and GM’s will remain out of sync for the duration of the TEK lifetime.

The command to trigger a key server role change is as below (to be executed on primary KS)

```
KS# clear crypto gdoi ks coop role
```

On execution of this command on a primary key server

- Key server relinquishes the primary role.
- DOES NOT clear the key server policies.
- Enters election process with other key servers.
- Key server with the highest priority will become primary.

If the priority values of the Key Servers have been changed prior to this, a new Primary KS will be selected as per the new priorities. Execution of this command on a secondary key server has no effect.

3.5.14 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 GETVPN 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.

3.5.14.1 Unicast Rekey

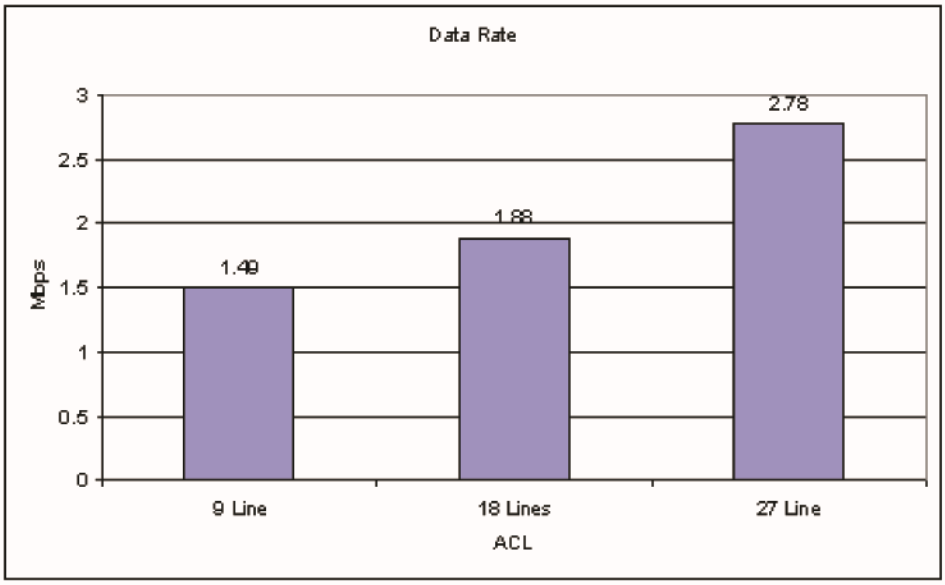
GETVPN rekey via unicast transport is recommended for deployment since it is more reliable method. The GM responds with a rekey acknowledgement and the KS processes the

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.

A KS with limited processing capabilities can also be a bottleneck in scaling to a very large GETVPNGETVPN (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 seconds). 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 GETVPN network.

Figure 7. Unicast Rekey Data Rate and ACL Size



3.5.14.2 Multicast Rekey

GETVPN 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.

For data integrity, SHA-256 (or higher) is recommended since it is more secure and stronger than SHA-1.

To configure:

```
crypto isakmp policy 10
 encr aes
 hash sha256
```

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.

Cisco recommends that the local deny policy be used for policy exceptions that are applicable only to a specific group member or for policy exceptions that are asymmetric. The global policy should be used for symmetric policy statements. If an asymmetric deny policy is applicable to every group member in the group, then it may be configured in the KS's global policy ACL.

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 172.16.4.2:
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

3.6.3.1 Difference in handling local deny policy between IOS and IOS XE based GMs

Unlike an IOS GM, the IOS XE GM does not reverse the source and destination IP addresses of an inbound packet before checking for a match in the GDOI ACL. As a result we will need an additional entry in the local deny ACL on the IOS XE GM that matches the inbound packet's source and destination ip address. Following is an illustration of the difference in behavior:

Consider the following local deny ACE:

"deny ip network-A network-B"

Following table shows the difference in behavior of IOS and IOS XE GMs for the above ACE.

	IOS-GM	IOS XE-GM
Send A to B in clear	Yes	Yes

Accept B to A in clear	Yes	No
Send B to A in clear	No	No
Accept A to B in clear	No	Yes

Consistent behavior between the 2 platforms can be achieved in either one of the following ways:
- Configure an extra deny entry in the IOS XE GM from network-B to network-A as follows
"deny ip network-B network-A"

OR

- Configure the following command on the IOS XE GM. This
"platform ipsec gdoi accept-both"

3.6.4 ACL for Fail-Closed Encryption Mode

A group member may operate in one of two modes prior to successfully registering. A group member's default mode is fail-open prior to registration. In this mode, the GM will allow all traffic to be forwarded regardless of the key server's crypto policy because the GM is not aware of the group's security policy. Once a group member registers, it operates in a fail-closed mode. In this mode, traffic that matches the downloaded policy must be encrypted or decrypted with the appropriate keys, or dropped if no keys exist, while traffic that does not match the policy may be forwarded in clear-text.

Many customers require the GM to operate in a fail-closed state both before and after registration. The feature allows the operator to define a local crypto policy ACL on the GM that places the GM in fail-closed mode all the times.

- If the GM has never registered or has its policy cleared using the 'clear crypto gdoi' command, it will operate in a fail-closed mode.
- If the GM successfully registers and receives rekeys, it will encrypt and decrypt the appropriate traffic according to the policy. Likewise, it will forward traffic that does not match the policy.
- If the GM does not receive a rekey and does not successfully re-register, it will drop traffic that matches the policy while continuing to forward traffic in clear-text that does not match the policy.

The GM always operates in fail-closed mode following a registration of any group configured on the GM.

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.

Fail-Close increases the security of GETVPN and the group member by enforcing that:

- Prior to registration or during registration, group member will drop any packets that arrive in the clear.
- Failure in any step during the registration process should also cause the group member to drop any packets in the clear.

The Fail-Close actions on the group member are relevant only for first time registration of the group member or if all policy is deleted using the “clear crypto gdoi” command on the group member. After the GM successfully registers to the KS, GM will keep that downloaded policy even if future re-registrations fail and IPSec SAs have expired. If Fail-Close is not configured on the group member, Fail-Open is the default behavior.

With both Fail-Close and Fail-Open modes, the GM keeps the policy it received from previous successful registration.

For Fail-Close mode to work correctly, a “complete” fail-close ACL has to be configured with all necessary deny ACEs for cleartext traffic. An example of an incomplete configuration is shown below:

```
ip access-list extended ACL_FC          <=== Incomplete ACL with no ACEs

ip access-list extended ACL_L2L

 deny   ip host 192.168.1.1 host 10.1.1.1

 permit ip any any

…….

crypto map 10 gdoi fail-close

 ! Incomplete                           <====

 ! Match address is not configured

 match address ACL_FC
```

An example of a complete configuration is shown below

```
       crypto ipsec transform-set tset1 esp-aes esp-sha256-hmac

       crypto map TEST gdoi fail-close

            match address 110

            activate

       crypto map TEST 10 ipsec-isakmp

            match address 120

            set peer 10.1.1.2

            set transform-set tset1

       crypto map TEST 20 gdoi

           match address 130

           set group ks1_group
```

In the above example, the Fail-close ACL is always matched LAST i.e. the order is as follows:
ACL 120 →ACL 130→ACL 110

All traffic that does not match any of the ACLs will be dropped. ACL 110 will add an implicit “Permit ip any any” This implicit entry is added because the crypto map is configured in Fail-Close mode. Permit means traffic will be dropped and deny means traffic will go in clear.

3.6.5 Group Member Re-registration

GM starts the re-registration process to the KS when it does not receive a rekey message from KS after a certain time.

3.6.5.1 Group Member Re-registration Values

The new **re-registration** timer value is calculated as the **MAX (5% of TEK Lifetime, 60s)**, that is, if 5% of the TEK Lifetime is less than 60s, the minimum default value of 60s will be used. The jitter value is added as a random number: RND (-6s to 6s).

Example below shows the calculation for the re-registration timer for a large TEK lifetime of 3600 seconds:

TEK Lifetime(T) = 3600 seconds

Re-registration time value is MAX(5% of T = 180s, 60s) = 180s

Jitter value is RND (-6s to 6s)

Re-registration in initiated for GM between (T-180s-6s) to (T-180s+6s)

The figure below illustrates the timer value as calculated in the example above. The GM re-registration timer is shown in relation to the KS rekey time which is MAX(10% of T= 360s, 90s) = 360s and also the time at which the TEK on the GM expires.

Figure 8.

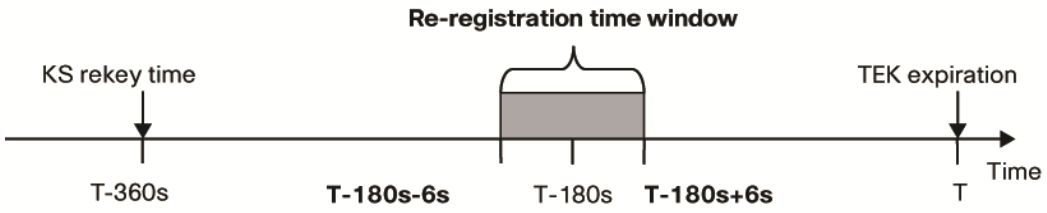


Figure 8 shows the reregistration timer values when calculated with an example of TEK Lifetime equal to 3600 seconds

Example below shows the calculation for the re-registration timer for a small TEK lifetime of 800 seconds:

TEK Lifetime(T) = 800s

Re-registration time value is **MAX(5% of T = 40s, 60s) = 60s**

Jitter value is RND(-6s to 6s)

Re-registration time window for GM is from (T-60s-6s) to (T-60s+6s)

The figure below illustrates the timer value as calculated in the example above. The GM re-registration timer is shown in relation to the KS rekey time which is MAX(10% of T = 80s, 90s) = 90s and also the time at which the TEK on the GM expires.

Figure 9.

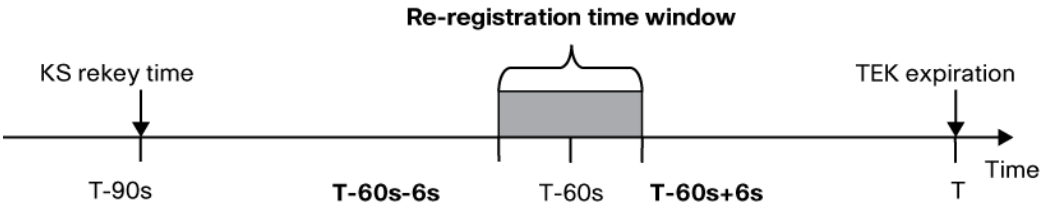


Figure 9 shows the reregistration timer values when calculated an example of TEK Lifetime equal to 800 seconds

3.6.5.2 Group Member Re-registration Jitter Values

In addition to the change in reregistration value, the new jitter value is calculated as a random value of **MAX(2% of TEK Lifetime, 6s)**, that is, if 2% of the TEK Lifetime is less than 6s, the minimum default value of 6s will be used. The re-registration time window becomes T-(5%+2%) to T-5%, which is **(T-7%) to (T-5%)**, where 5% is the new re-registration time illustrated in section 3.6.5.1 above. The re-registration timer value remains MAX(5% of TEK Lifetime, 60s).

Example below shows the calculation for the re-registration timer and the jitter value for a large TEK lifetime of 3600 seconds:

TEK Lifetime(T) = 3600s

Re-registration time value is MAX(5% of T = 180s, 60s) = 180s

Jitter value is MAX(2% of T = 72s, 6s) = 72s

Re-registration in initiated for GM between (T-180s-72s) to (T-180s)

The figure below illustrates the re-registration timer value with jitter as calculated in the example above. The GM re-registration timer is shown in relation to the KS rekey time which is MAX(10% of T= 360s, 90s) = 360s and also the time at which the TEK on the GM expires.

Figure 10.

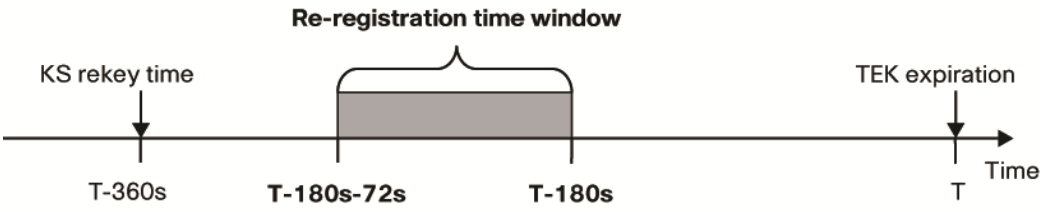


Figure 10 shows the re-registration timer values with jitter for a TEK Lifetime equal to 3600 seconds

Example below shows the calculation for the re-registration timer for a small TEK lifetime of 200 seconds:

TEK Lifetime(T) = 200s

Re-registration time value is MAX(5% of T = 10s, 60s) = 60s

Jitter value is MAX(2% of T = 4s, 6s) = 6s

Re-registration is initiated for GM between (T-60s-6s) to (T-60s)

The figure below illustrates the timer value as calculated in the example above. The GM re-registration timer is shown in relation to the KS rekey time which is MAX(10% of T = 20s, 90s) = 90s and also the time at which the TEK on the GM expires.

Figure 11.

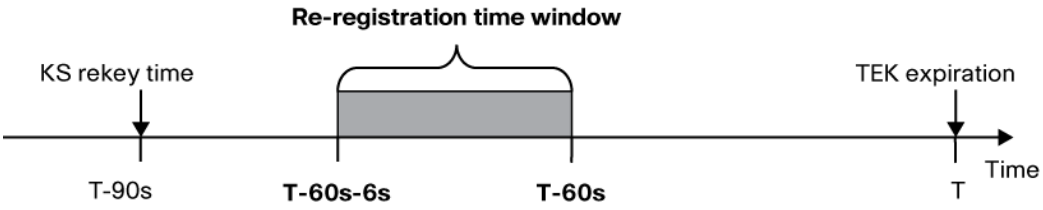


Figure 11 shows the re-registration timer values with jitter for a TEK Lifetime equal to 200 seconds

Note: When a GM with ISM-VPN module boots up, it is possible that the GM may first register and download the policy to the onboard crypto engine and then re-register again to download the policy onto the ISM-VPN when the module boots up.

3.6.5.3 Re-registration Differences Between G-IKEv2 and GDOI

- IKEv1 Retransmission for GDOI Registration Failure
  - IKEv1 uses constant retransmission interval (3x 10 sec + 10 sec backoff)
  - GDOI uses exponential backoff (70, 140, 280, 480)
- IKEv2 Retransmission for G-IKEv2 Registration Failure
  - IKEv2 uses an exponential backoff for retransmission (2, 4, 8, 16, ...)
  - G-IKEv2 caps retransmission at 4 (2+4+8+16 = 30 sec + 10 sec backoff)
  - G-IKEv2 uses same backoff (70, 140, 280, 480) at end of KS list

3.6.6 GETVPN KEK Rekey Behavior Changes

This section describes GETVPN KEK Encryption Key (KEK) rekey behavior changes in Cisco IOS Release 15.2(1)T) and Cisco IOS XE Release 3.5S/15.2(1)S) (corresponding GDOI version). Prior to these releases, the KEK rekey is sent by the Key Server (KS) when a current KEK expires. The Group Member (GM) does not maintain a timer to keep track of the remaining lifetime of the KEK. The current KEK is replaced by a new KEK only when a KEK rekey is received. If the GM does not receive a KEK rekey at the expected KEK expiry, it does not trigger a reregistration to the KS, and it will retains the existing KEK without allowing the KEK to expire. This may result in the KEK being used after its configured lifetime. There is no command that can be used on the GM that displays the remaining lifetime of the KEK.

3.6.6.1 Key Server Behavior

Effective with Cisco IOS Release 15.2(1)T and Cisco IOS XE Release 3.5S/15.2(1)S – (corresponding GDOI version, the new KEK rekey behavior includes the following changes:

- On the KS - KEK rekeys are sent before the current KEK expiry, much like a Traffic Exchange Key (TEK) rekey.
- On the GM - The GM maintains a timer to keep track of the remaining KEK lifetime and triggers a reregistration if the KEK rekey is not received.

With the new rekey behavior, the KS starts a KEK rekey before the current KEK expiry according to the below formula.

KEK rekey time = KEK lifetime – (200 + (number of retransmissions x retransmission interval) + (5 x (1 + Number of registered GMs/50))

Note: In the above formula, the 5 x (1 + Number of registered GMs/50) portion is used with a unicast rekey.

Per this behavior, a KS starts to rekey a KEK 200 seconds before the current KEK expires. After the rekey is sent, the KS starts to use the new KEK for all subsequent TEK/KEK rekeys.

3.6.6.2 Group Member Behavior

Effective with Cisco IOS Release 15.2(1)T and Cisco IOS-XE Release 3.5S/15.2(1)S), the new GM behavior includes the following changes:

1. GM enforces a KEK lifetime expiry by adding a timer to keep track of the KEK remaining lifetime. When that timer expires, the KEK is deleted on the GM and a reregistration is triggered.
2. GM expects a KEK rekey to occur 200 seconds before the current KEK expires. A timer is added so that so that a KEK is deleted and a re-registration is triggered when a new KEK is not received 200 seconds before the current KEK expires. This KEK deletion and reregistration event happens in the timer interval of (KEK expiry - 190 seconds, KEK expiry - 40 seconds).

Along with the functional changes, the GM show command outputs are also modified to display the KEK remaining lifetime accordingly.

```
GM#show                              crypto                              gdoi
GROUP                                                                INFORMATION

    Group    Name                                                        :    G1
    Group     Identity                                                  :     3333
    Crypto    Path                                                      :    ipv4
    Key      Management      Path                                      :      ipv4
    Rekeys     received                                                 :     0
    IPSec      SA      Direction                                       :      Both

     Group     Server     list                                  :     10.1.11.2

    Group  member                        :  10.1.13.2               vrf:  None
       Version                                                       :    1.0.4
       Registration         status                          :         Registered
       Registered     with                                      :     10.1.11.2
       Reregisters in : 81 sec       <=== Reregistration due to TEK or KEK,
whichever                           comes                                first
       Succeeded                      registration:                        1
       Attempted                      registration:                        1
       Last     rekey     from                                  :     0.0.0.0
       Last       rekey       seq       num                          :       0
```

```
        Unicast                 rekey                 received:                0
        Rekey      ACKs      sent                                      :      0
        Rekey     Received                                           :      never
        allowable                rekey                cipher:                any
        allowable          rekey          hash                    :          any
        allowable               transformtag:               any               ESP

     Rekeys                                                                cumulative
        Total      received                                                 :      0
        After             latest             register             :             0
        Rekey      Acks      sents                                      :      0

 ACL            Downloaded            From            KS            10.1.11.2:
   access-list                                  deny           ospf           any
   access-list                           deny         eigrp         any         any
   access-list         deny   udp   any   port   =   848   any   port   =   848
   access-list                           deny         icmp         any         any
   access-list                           permit         ip         any         any

KEK                                                                        POLICY:
    Rekey      Transport      Type                                  :      Unicast
    Lifetime (secs) : 56       <=== Running timer for remaining KEK lifetime
    Encrypt      Algorithm                                          :      3DES
    Key    Size                                                     :    192
    Sig     Hash     Algorithm                                :     HMAC_AUTH_SHA
    Sig       Key       Length       (bits)                        :       1024

TEK     POLICY     for     the     current     KS-Policy     ACEs     Downloaded:
  Serial1/0:
    IPsec                                                                  SA:
        spi:                                          0xD835DB99(3627408281)
        transform:                    esp-3des                    esp-sha-hmac
        sa      timing:remaining      key      lifetime      (sec):      (2228)
        Anti-Replay(Time Based) : 10 sec interval
```

3.6.6.3 Interoperability Issues

To avoid interoperability issues, it is recommended that you upgrade the GM and the KS to one of the versions supporting the new KEK rekey behavior.

If the GM is running the older code, and the KS is running the software version, the KS sends out the KEK rekey prior to the KEK expiry and there is no additional functional impact. However, if a GM running the newer code registers with a KS running the older code, the GM may incur two Group Domain of

Interpretation (GDOI) reregistrations to receive the new KEK per KEK rekey cycle. The following events occur when this happens:

1. The GM reregisters before the current KEK expiry, since the KS will only send the KEK rekey when the current KEK expires. The GM receives the KEK, and it is the same KEK as the one it currently has with less than 190 seconds lifetime remaining. This tells the GM that it is registered with a KS without the KEK rekey change.

```
%GDOI-4-GM_RE_REGISTER: The IPSec SA created for group G1 may
   have expired/been cleared, or didn't go through. Re-register to KS.
%CRYPTO-5-GM_REGSTER: Start registration to KS 10.1.11.2 for
   group G1 using address 10.1.13.2
%GDOI-5-GM_REKEY_TRANS_2_UNI: Group G1 transitioned to Unicast Rekey.
%GDOI-5-SA_KEK_UPDATED: SA KEK was updated
%GDOI-5-SA_TEK_UPDATED: SA TEK was updated
%GDOI-5-GM_REGS_COMPL: Registration to KS 10.1.11.2 complete
   for group G1 using address 10.1.13.2
%GDOI-5-GM_INSTALL_POLICIES_SUCCESS: SUCCESS: Installation of
   Reg/Rekey policies from KS 10.1.11.2 for group G1 & gm identity
  10.1.13.2
```

2. The GM deletes the KEK at its lifetime expiry, and sets a reregistration timer of (KEK expiry, KEK expiry + 80).

```
%GDOI-5-GM_DELETE_EXPIRED_KEK: KEK expired for group G1 and was deleted
```

3. When the reregistration timer expires, the GM reregisters and will receive the new KEK.

```
%GDOI-4-GM_RE_REGISTER: The IPSec SA created for group G1 may
   have expired/been cleared, or didn't go through. Re-register to KS.
%CRYPTO-5-GM_REGSTER: Start registration to KS 10.1.11.2 for
   group G1 using address 10.1.13.2
%GDOI-5-GM_REKEY_TRANS_2_UNI: Group G1 transitioned to Unicast Rekey.
%GDOI-5-SA_KEK_UPDATED: SA KEK was updated
%GDOI-5-SA_TEK_UPDATED: SA TEK was updated
%GDOI-5-GM_REGS_COMPL: Registration to KS 10.1.11.2 complete for
   group G1 using address 10.1.13.2
%GDOI-5-GM_INSTALL_POLICIES_SUCCESS: SUCCESS: Installation of
   Reg/Rekey policies from KS 10.1.11.2 for group G1 & gm identity 10.1.13.2
```

3.6.6.4 GETVPN Resiliency - GM Error Detection

The GETVPN Resiliency - GM Error Detection/Error Recovery feature detects erroneous packets in the data plane for each Group Domain of Interpretation (GDOI) group such as invalid SPIs or Time-Based Anti-Replay (TBAR) errors. These errors are tracked, and the outer source IP address of the packet is recorded. The KS encodes the group information in the SPI (Security Parameter Index) and then it downloads it via the TEK policy to the GM.

The following configuration snippet shows how to enable the group member (GM) to monitor for control-plane errors every 300 seconds.

```
crypto gdoi group GETVPN
 identity number 1111
 server address ipv4 1.0.0.2
 client recovery-check interval 300
```

When a failure is detected by the GETVPN Resiliency - GM Error Detection feature, a syslog message is generated to show the source IP address of the erroneous packet:

```
%GDOI-4-TIMEBASED_REPLAY_FAILED: An anti replay check has failed in group GETVPN from sourceip-address 100.0.0.9. my_pseudotime is 600006.78 secs,peer_pseudotime is 500033.34 secs, replay_window is 100(second)

*Feb 10 21:01:56.043:%CRYPTO-4-PKT_REPLAY_ERR: decrypt: replay check failed connection id=29, sequence number=11
```

A syslog message is generated to notify that this GM recovery reregistration feature is triggered. For instance, if you configure the GM to monitor for control-plane errors every 300 seconds, when the recovery registration occurs the following syslog is generated:

```
*Feb 23 19:06:28.600: %GDOI-5-GM_RECOVERY_REGISTER: received invalid GDOI packets; register to KS to refresh policy, keys, and PST.
```

Note: Error Detection is not supported on ASR1000 GM.

For more information on this feature, refer to GETVPN Resiliency - GM Error Detection feature.

3.7 COOP Design Considerations

COOP KSs provide redundancy to GETVPN. Multiple KSs are supported by GETVPN 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 distribute 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.

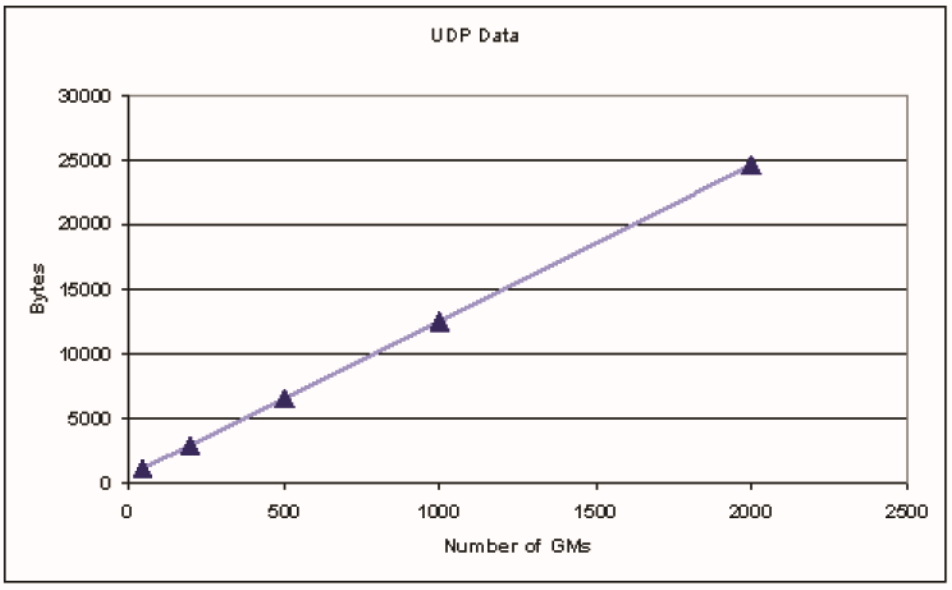
3.7.1.1 COOP Message Size

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 3.7.3.8 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 12.

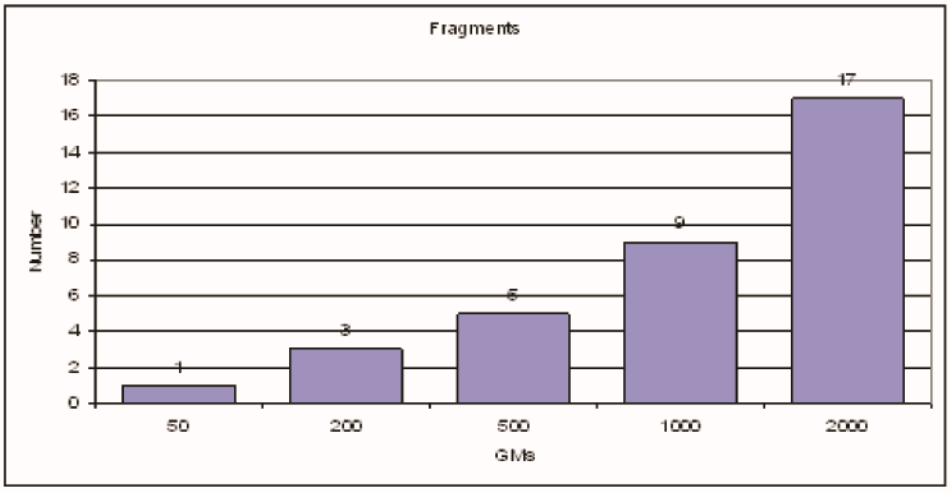
Figure 12. COOP Message Size by Number of GMs



For a fixed size group policy configured on the KS (nine lines of ACL — eight denies and one permit), Figure 11 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 13 shows how fragments increase with the increase in number of GMs. For 2000 GMs, for example, the message is contained in 17 fragments.

Figure 13. 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 GETVPN 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: 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: 172.18.5.2
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: 172.20.4.2
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 snippet:

```
KS2#sh cry gdoi ks coop
<..>
Peer Sessions:
Session 1:
Server handle: 2147483651
Peer Address: 172.16.4.2
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: 172.20.4.2
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.

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. It is imperative that the same GETVPN policies be configured on all of the COOP KS serving a group.

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>
!
```

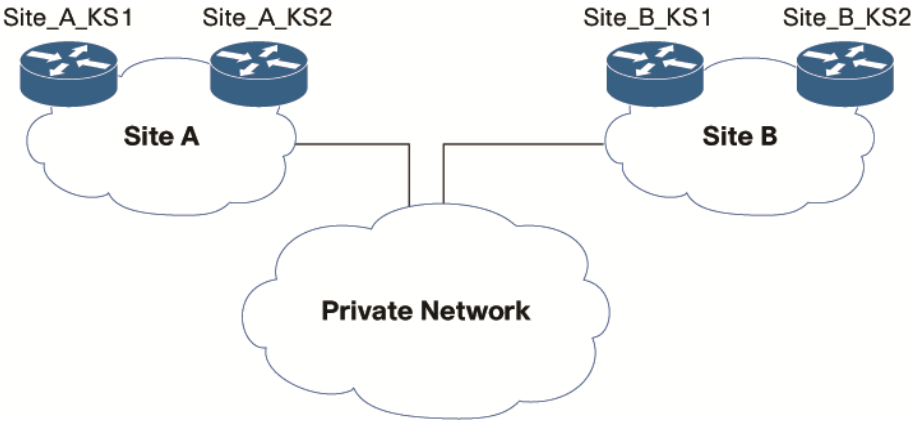
3.7.3.2 Setting KS Priority

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 14. Assume that each site has two KSs, so four KSs operate cooperatively in the same group.

Figure 14. 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>
 identity number 12345
 server local
  address ipv4 172.16.4.2
  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)

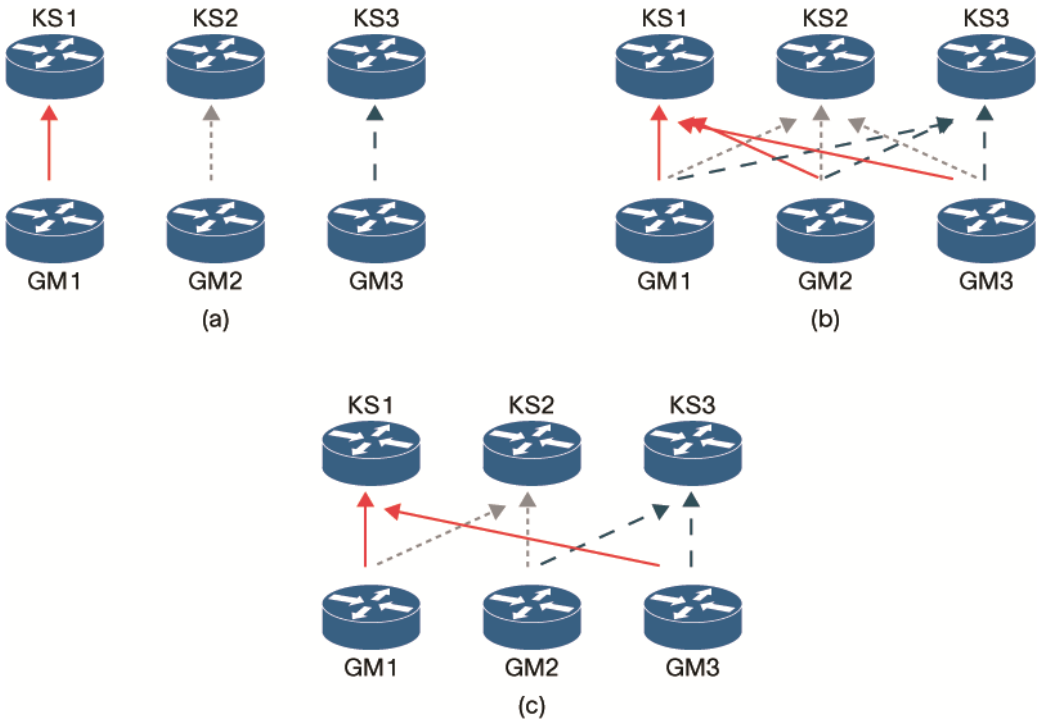
3.7.3.3.1 Load Balancing using Configuration

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 15 (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 15 (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 15 (c).

Figure 15. 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>
!
```

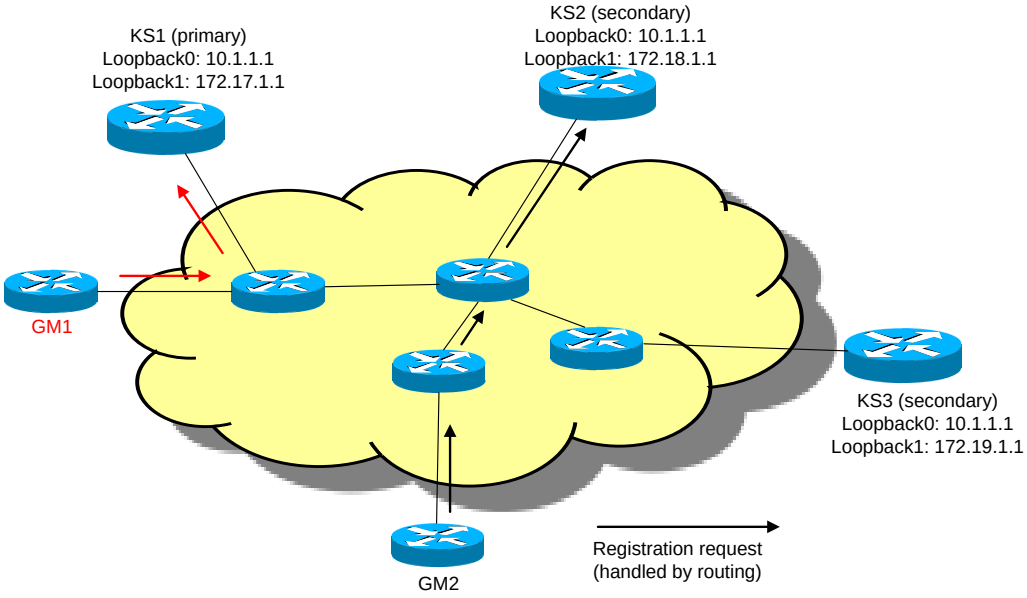
3.7.3.3.2 Load Balancing using Routing (Anycast)

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 14 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 16 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 16. Load Balancing GM Registrations using Routing



On the GMs, only one KS must be configured.

```
!
crypto gdoi group GETVPN
```

```
 server address ipv4 10.1.1.1
!
```

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 16.

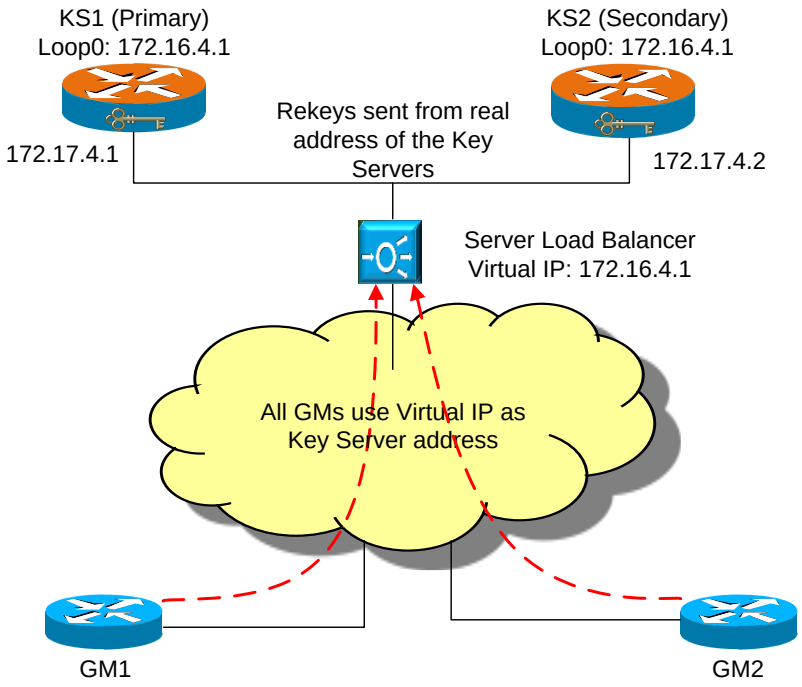
```
!
! Common IP address
interface Loopback0
 ip address 10.1.1.1 255.255.255.255
!
! Distinct IP address
interface Loopback1
 ip address 172.17.1 255.255.255.255
!
crypto gdoi group GETVPN
 identity 12345
 server local
  registration interface Loopback0
  address ipv4 172.17.1.1
  redundancy
   local priority 255
   peer address ipv4 172.18.1.1
   peer address ipv4 172.19.1.1
!
```

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 17 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 17. Load Balancing GM Registrations Using SLB



On the GMs, only one KS must be configured.

```
!
crypto gdoi group GETVPN
server address ipv4 172.16.4.1
!
```

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 172.17.4.1
  weight 1
  maxconns 500
  inservice
 !
 !KS-2
 real 172.17.4.2
  weight 1
  maxconns 500
  inservice
!
!define a virtual server for GDOI protocol (UDP 848)
ip slb vserver GDOI
 virtual 172.16.4.1 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 17.

```
!
crypto gdoi group GETVPN
server local
 address ipv4 172.17.4.1
 redundancy
  peer address ipv4 172.17.4.2
!
```

3.7.3.4 Timers for registration failover to alternate Key Servers

When a GM is configured with more multiple KSs in the group, it will try to register with the first KS configured and will move to the subsequent KSs in in case the registration fails. The GM will wait for a specified time before trying to register wit h the next KS. This wait time (also known as IKE backoff timer) is

40seconds till the GM reaches the last configured KS. In case the registration is not successful with any of the KSs in the first pass, the GM will retry the registration process with the first KS after waiting for 70seconds. The wait time between passes is known as the GDOI backoff timer.

In the second pass, the GM will try to register with the KSs in the order of configuration. Similar to the first pass, the GM will wait for 40seconds between registrations. However, at the end of the second pass, the GDOI backoff timer will now be 140 seconds (i.e., the GM will now wait for 140 seconds before restarting the third pass).

The GM will keep trying to register with the KSs till a successful registration. The GDOI backoff timer will keep doubling after every pass till it reaches a maximum value of 480seconds. The following table summarizes the registration failover timers.

Let us assume the GM is configured with 3 KSs (KS1, KS2 and KS3). The wait times between registrations are shown below:

Pass	Registration Failure #	Failed to register with KS	Wait time (seconds)	Timer type
1	1	KS1	40	IKE backoff
	2	KS2	40	IKE backoff
	3	KS3	70	GDOI backoff
2	4	KS1	40	IKE backoff
	5	KS2	40	IKE backoff
	6	KS3	140	GDOI backoff
3	7	KS1	40	IKE backoff
	8	KS2	40	IKE backoff
	9	KS3	280	GDOI backoff
4	10	KS1	40	IKE backoff
	11	KS2	40	IKE backoff
	12	KS3	480	GDOI backoff <== Max backoff value
5 (and higher)	13	KS1	40	IKE backoff
	14	KS2	40	IKE backoff
	15	KS3	480	GDOI backoff <== Max backoff value

3.7.3.5 Identical Policies on all KSs

With multiple COOP KSs, the policies configured on each KS must be considered. **The same GETVPNGETVPN policies should be configured on each of the Key Servers.**

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, policy changes 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.

3.7.3.6 RSA Key on COOP KSs

KSs require RSA keys to create KEK keys. In GETVPN, 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.

3.7.3.7 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.

3.7.3.8 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
!
```

3.7.3.9 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. Therefore, the above rekey occurs only at TEK expiry 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.

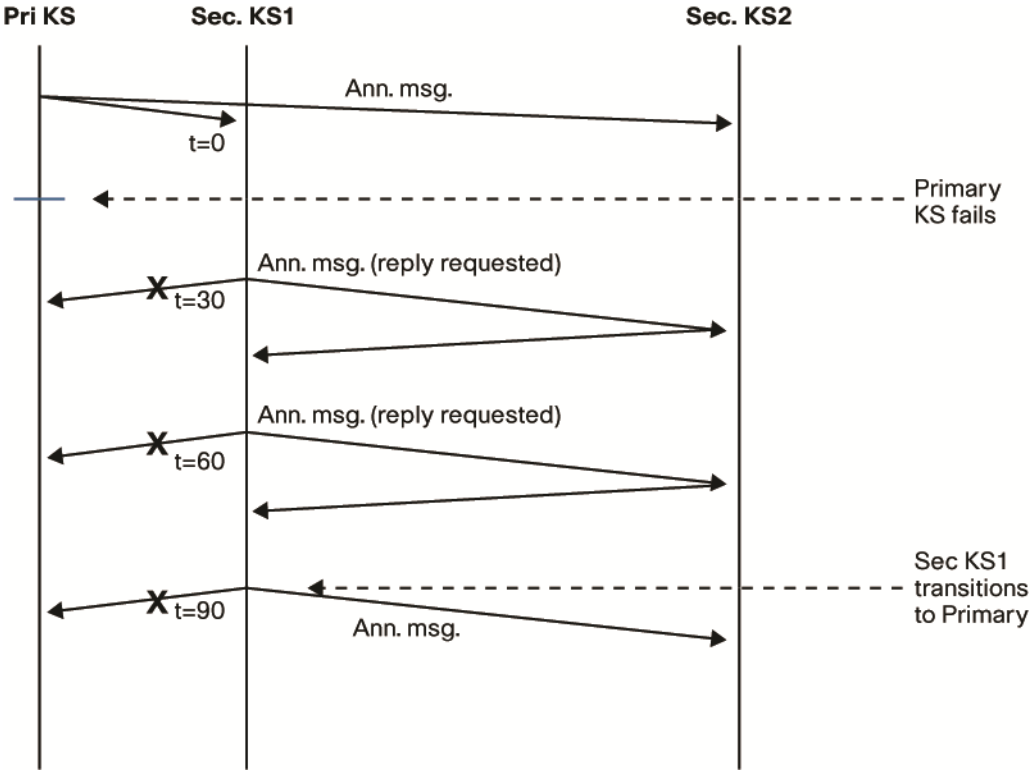
3.7.4.1 KS Failure

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 18 shows secondary KS behavior just before it transitions to the primary state.

Figure 18. 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: 172.16.4.1

Local Priority: 100

Local KS Role: Secondary , Local KS Status: Alive

<..>

Peer Sessions:

Session 1:

Server handle: 2147483651

Peer Address: 172.16.4.3

Peer Priority: 1

Peer KS Role: Secondary, Peer KS Status: Unknown

<..>

Session 2:

Server handle: 2147483656
```

```
Peer Address: 172.16.4.2

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 172.16.4.1 in group
dgvpn1 transitioned to Primary (Previous Primary = 172.16.4.2)


May 7 10:21:30.328: %GDOI-3-COOP_KS_UNREACH: Cooperative KS 172.16.4.2
Unreachable in group dgvpn1


May 7 10:21:32.628: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for
group dgvpn1 from address 172.16.4.1 with seq # 1


May 7 10:21:50.329: %GDOI-3-COOP_KS_UNREACH: Cooperative KS 172.16.4.2
Unreachable in group dgvpn1


May 7 10:22:10.330: %GDOI-3-COOP_KS_UNREACH: Cooperative KS 172.16.4.2
Unreachable in group dgvpn1


May 7 10:22:30.331: %GDOI-3-COOP_KS_UNREACH: Cooperative KS 172.16.4.2
Unreachable in group dgvpn1
```

The **COOP_KS_TRANS_TO_PRI** message indicates a change in primary KS status: KS 172.16.4.1 is the new primary KS; the previous primary KS was KS 172.16.4.2.

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 172.16.4.2. Note that these messages are 20 seconds apart (the policy update timer period).

After transitioning to primary, Sec-KS1 now shows Sec-KS2 (172.16.4.3) as Secondary/Alive and shows Pri-KS as Primary/Dead.

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


<..>
```

```
Local Address: 172.16.4.1
Local Priority: 100
Local KS Role: Primary , Local KS Status: Alive
<..>

Peer Sessions:
Session 1:
Server handle: 2147483651
Peer Address: 172.16.4.3
Peer Priority: 50
Peer KS Role: Secondary , Peer KS Status: Alive
<..>

Session 2:
Server handle: 2147483656
Peer Address: 172.16.4.2
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 172.16.4.2 (Pri-KS) restored.

```
Sec-KS1#

May 7 10:51:30.473: %GDOI-5-COOP_KS_REACH: Reachability restored with

Cooperative KS 172.16.4.2 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.

The new primary KS now preserves the existing KEK/TEK during split which mitigates the necessity of rekeying immediately after a split. It also mitigates the necessity of a rekey during a merge that may immediately follow due to route re-convergence.

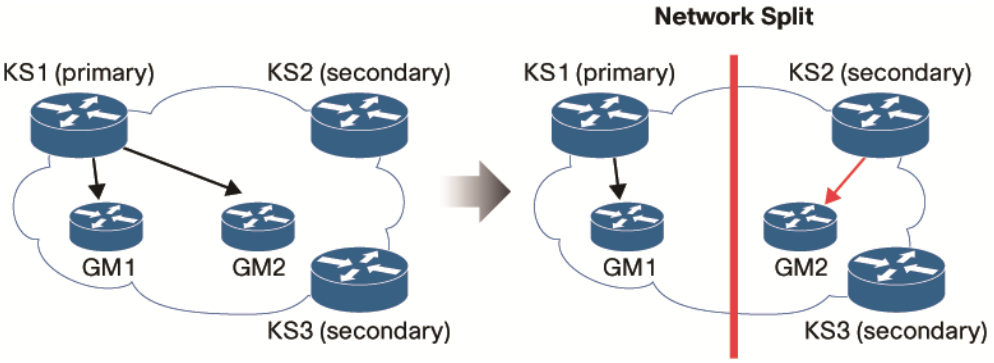
3.7.4.2.1 Network Split and Merge (KS and GM Split)

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 19 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 19. 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. In the figure above, KS2 only creates a new KEK/TEK if the old one is about to expire.
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: 172.16.4.2

Local Priority: 175

<..>

Local KS Role: Primary , Local KS Status: Alive

<..>

Peer Address: 172.16.4.3      <-- KS3 status

Peer Priority: 50
```

```
Peer KS Role: Secondary , Peer KS Status: Alive
<..>
Peer Address: 172.16.4.1      <-- KS2 status
Peer Priority: 100
Peer KS Role: Secondary , Peer KS Status: Alive

!KS2
KS2#sh cry gdoi ks coop
<..>
Local Address: 172.16.4.1
Local Priority: 100
Local KS Role: Secondary , Local KS Status: Alive
<..>
Peer Address: 172.16.4.3      <-- KS3 status
Peer Priority: 50
Peer KS Role: Secondary , Peer KS Status: Unknown
<..>
Peer Address: 172.16.4.2      <-- 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 :      192.168.100.1 172.16.4.1    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 :     192.168.1.1   172.16.4.1    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 172.16.4.1 in group
dgvpn1 transitioned to Primary (Previous Primary = 172.16.4.2)

!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 172.16.4.1 with seq # 1

!message indicating that KS2 can't reach KS1
May 8 09:39:22.960: %GDOI-3-COOP_KS_UNREACH: Cooperative KS 172.16.4.2
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: 172.16.4.2
Local Priority: 175
Local KS Role: Primary , Local KS Status: Alive
<..>
Peer Address: 172.16.4.3      <-- KS3 status
Peer Priority: 50
Peer KS Role: Secondary, Peer KS Status: Dead
<..>
```

```
Peer Address: 172.16.4.1      <-- KS2 status
Peer Priority: 100
Peer KS Role: Secondary, Peer KS Status: Dead

!KS2
KS2#show cry gdoi ks coop
<..>
Local Address: 172.16.4.1
Local Priority: 100
Local KS Role: Primary , Local KS Status: Alive
<..>
Peer Address: 172.16.4.3      <-- KS3 status
Peer Priority: 50
Peer KS Role: Secondary , Peer KS Status: Alive
<..>
Peer Address: 172.16.4.2      <-- 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 (192.168.1.1) received rekey from
!KS2(172.16.4.1). Rekey sequence number is 1

May 8 10:39:18.054: %GDOI-5-GM_RECV_REKEY: Received Rekey for group dgvpn
from 172.16.4.1 to 192.168.1.1 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: ACTIVEAfter 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#show crypto gdoi ks member | begin 192.168.1.1      <--    check    GM2
status
Group Member ID  : 192.168.1.1
Group ID  : 61440
Group Name       : dgvpn1
Key Server ID    : 172.16.4.1
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
```

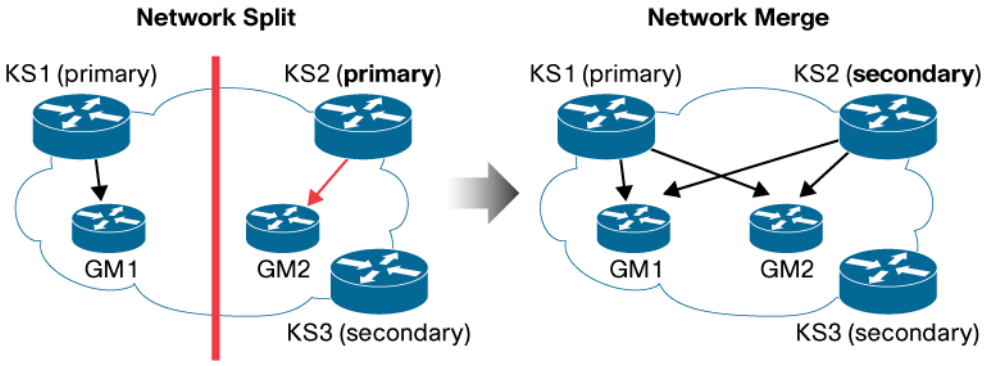
After the third rekey, that is, just before the fourth rekey is sent, KS1 deletes GM2.
Similarly, KS2 deletes GM1.
The following log messages from KS1 and KS2 indicate that the GM is deleted.

```
!From KS1
May 8 11:30:28.026: %GDOI-4-GM_DELETE: GM 192.168.1.1 deleted from group
dgvpn1.


!From KS2
```

May 8 11:30:08.157: %**GDOI-4-GM_DELETE:** GM 192.168.100.1 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 20.

Figure 20. 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 172.16.4.3 in group dgvpn1.

May 8 11:36:48.066: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for
group dgvpn1 from address 172.16.4.2 with seq # 11


!messages from KS2
May 8 10:36:38.213: %GDOI-5-COOP_KS_REACH: Reachability restored with
Cooperative KS 172.16.4.2 in group dgvpn1.

May 8 10:36:48.033: %GDOI-5-COOP_KS_TRANS_TO_PRI: KS 172.16.4.2 in group
dgvpn1 transitioned to Primary (Previous Primary = 172.16.4.1)

May 8 10:36:48.065: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for
group dgvpn1 from address 172.16.4.1 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: 172.16.4.1
Local Priority: 100
Local KS Role: Secondary , Local KS Status: Alive
<..>
Peer Address: 172.16.4.3      <-- KS3 status
Peer Priority: 50
Peer KS Role: Secondary , Peer KS Status: Unknown
<..>
Peer Address: 172.16.4.2      <-- 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 172.16.4.2 and another under server 172.16.4.1 (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 172.16.4.2 (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 172.16.4.1 (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 (172.16.4.2)
May 8 11:36:48.120: %GDOI-5-GM_RECV_REKEY: Received Rekey for group dgvpn
```

```
from 172.16.4.2 to 192.168.100.1 with seq # 11

!GM2

!rekey received from KS2 (172.16.4.1)
May 8 11:36:48.131: %GDOI-5-GM_RECV_REKEY: Received Rekey for group dgvpn
from 172.16.4.1 to 192.168.1.1 with seq # 11

!rekey received from KS1 (172.16.4.2)
May 8 11:36:48.211: %GDOI-5-GM_RECV_REKEY: Received Rekey for group dgvpn
from 172.16.4.2 to 192.168.1.1 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
```

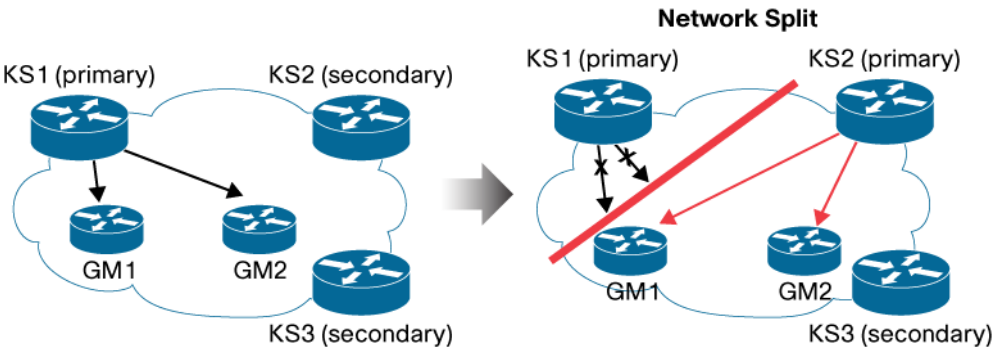
To summarize: During this kind of network split, GMs have connectivity to the KSs in their respective partition, but do not have connectivity across partitions. After the network split is resolved, the primary KS sends out multiple rekeys, each encrypted using one of the different KEKs, so that all GMs 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 21, 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 21. Network Split and Merge: Only KS Split



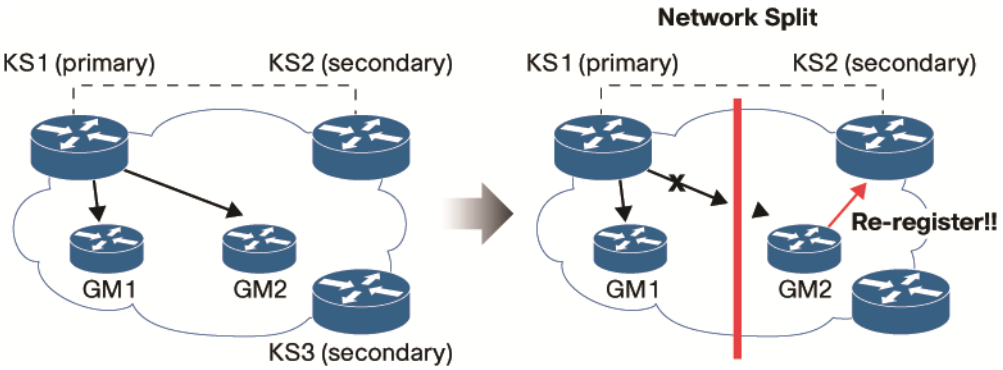
This scenario is almost the same as that described in 3.7.4.1 “KS Failure.” The main difference in this scenario is that during the split, KS1 and KS2 both become primary KSs, but that rekeys from KS1 never makes it to any GM. When the network merge occurs, KS2 sees another higher priority KS, and so 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 22 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 22. 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 GETVPNGETVPN 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 GETVPN 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 Hosts 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 GETVPN. Refer to CSCsq23600 for more details

3.8.2 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 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.

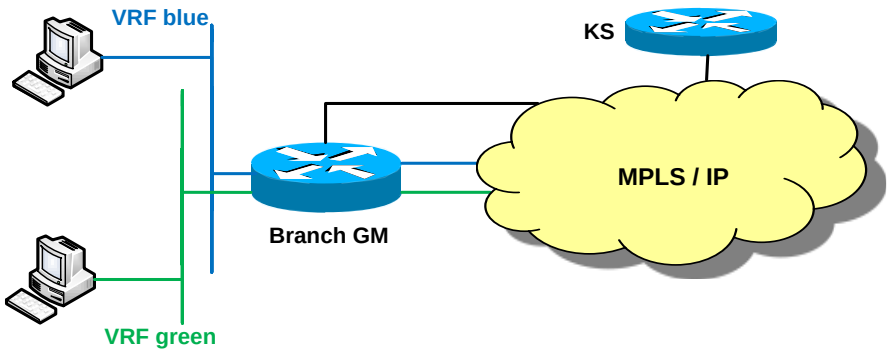
If this traffic is TCP based, tcp adjust-mss will mitigate the problem. If the traffic is not TCP based, as a last resort df-bit can be cleared on the GM using crypto ipsec df-bit clear in **global** configuration mode. This will result in clearing of the df-bit and packet will be fragmented. Both the 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.

3.9 VRF-Aware GETVPN

In certain scenarios, it might be desirable to deploy VRFs in GETVPN solution. It must be noted that GETVPN GM is VRF aware but the GETVPN KS is not. The VRF aware GM feature facilitates the separation of control traffic (registration and rekey) from data plane traffic. This is facilitated by the ability of the GMs to route control traffic (registration & rekeys) through a VRF, which is different from the data traffic. Basically registration and rekeys are routed through one VRF and the policies downloaded are applied in a different forwarding VRF. Note that only VRF lite is supported on data plane.

Consider the following deployment:

Figure 23. VRF-Aware GM



Each VRF will require a unique WAN interface/sub-interface to apply the crypto map

1. 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 172.16.1.1
 client registration interface Serial1/0/0
!
crypto gdoi group green
 identity number 102
 server address ipv4 172.17.1.1
!
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
 ip address 172.19.1.2 255.255.0.0
 crypto map blue
!
```

```
interface Serial0/0/0:1.102
 description WAN interface for VRF green
 ip vrf forwarding green
 ip address 172.20.1.2 255.255.0.0
 crypto map green
!
interface Serial1/0/0
 description getvpn control traffic interface
 ip address 172.18.1.2 255.255.0.0
!
```

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.

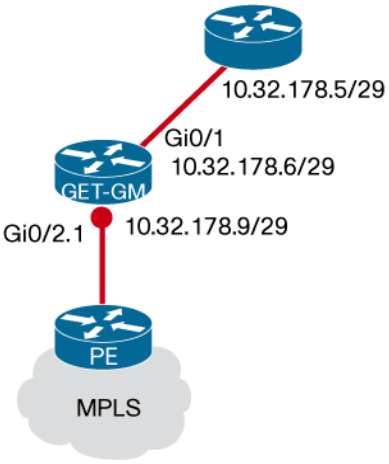
The best practice is to configure the KS set to use a designated route partition specifically for the GDOI management and control plane (for example, network operations center (NOC) routing segment). GM IKE identities can be advertised to the NOC routing segment so that the KS can reach all GM VRF subinterfaces. 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.

VRF_Lite and Route Leaking

GETVPN GM requires VRF-lite in data path. VRF-lite means that all the traffic traversing in the particular VRF should confide to the same VRF after route lookup. VRF-lite does not cover route leaking. If route leaking is configured on the GM, packets will be sent out in clear text from the crypto map applied interface.

In VRF-Aware GET-GM Group Member, all the route lookups for traffic coming in from interface GigabitEthernet0/1 and routed out through the interface GigabitEthernet0/2.1 will happen in VRF RED. For this case, GETVPN IPSec SAs will be installed in the VRF RED. Any traffic coming in the VRF RED and routed out the interface GigabitEthernet0/2.1 will be classified by the GETVPN crypto map getvpn-map policies.

Figure 24. Topology of VRF-lite data path



Example configuration of GET-GM:

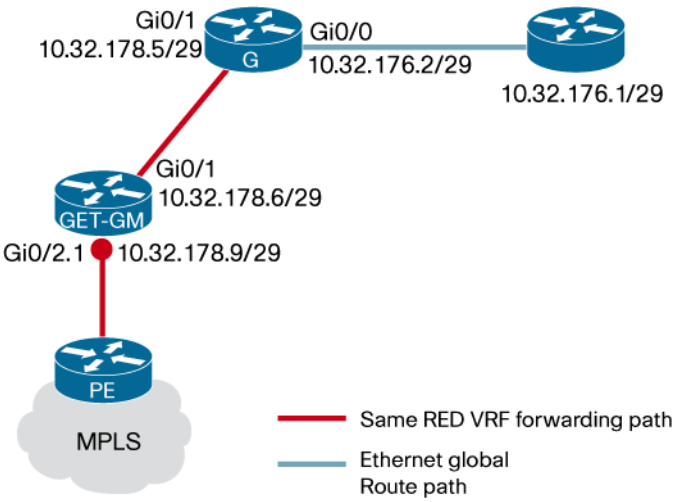
```
ip vrf RED
 rd 1:100
 route-target export 1:100
 route-target import 1:100
!
ip vrf management
 rd 1:299
 route-target export 1:299
 route-target import 1:299

crypto gdoi group GETVPN-RED
 identity number 1357924680
 server address ipv4 10.32.178.56
 server address ipv4 10.32.178.23
 client registration interface GigabitEthernet0/2.2

crypto map GETVPN-map 1 gdoi
 set group GETVPN-RED
 match address no-encryption-acl


interface GigabitEthernet0/1
```

```
 ip address 10.32.178.6 255.255.255.252
ip vrf forwarding RED
…

interface GigabitEthernet0/2
 no ip address
…
interface GigabitEthernet0/2.1
 description outside interface for data traffic
 encapsulation dot1Q 1033
 ip vrf forwarding RED
 ip address 10.32.178.9 255.255.255.252
crypto map GETVPN-map

interface GigabitEthernet0/2.2
description GETVPN control traffic interface
 encapsulation dot1Q 1066
 ip vrf forwarding management
 ip address 10.32.178.13 255.255.255.252
…
```

If route leaking is required to make the traffic flow from an interface participating in global routing to another interface with VRF forwarding or vice-versa, the following is required:

Enable route leaking on the router behind the GM and make sure the incoming traffic goes into the same VRF in the GM and routed out of the same VRF interface where crypto map is applied.

Figure 25. Topology with route leaking



Example route leaking configuration in G router behind the GM:

```
interface GigabitEthernet0/0
 ip address 10.32.176.2 255.255.255.252


interface GigabitEthernet0/1
 ip vrf forwarding RED
 ip address 10.32.178.5 255.255.255.252


! Packet routed from global routing table destined for RED VRF network
10.32.178.0/25,
!   inject the traffic into RED VRF path
ip route 10.32.178.0 255.255.255.0  GigabitEthernet0/1
! Return packet routed from RED VRF, destined for a network in global
routing table,
!  inject the traffic  into global routing path
ip route vrf RED 0.0.0.0 0.0.0.0 10.32.176.1 global
```

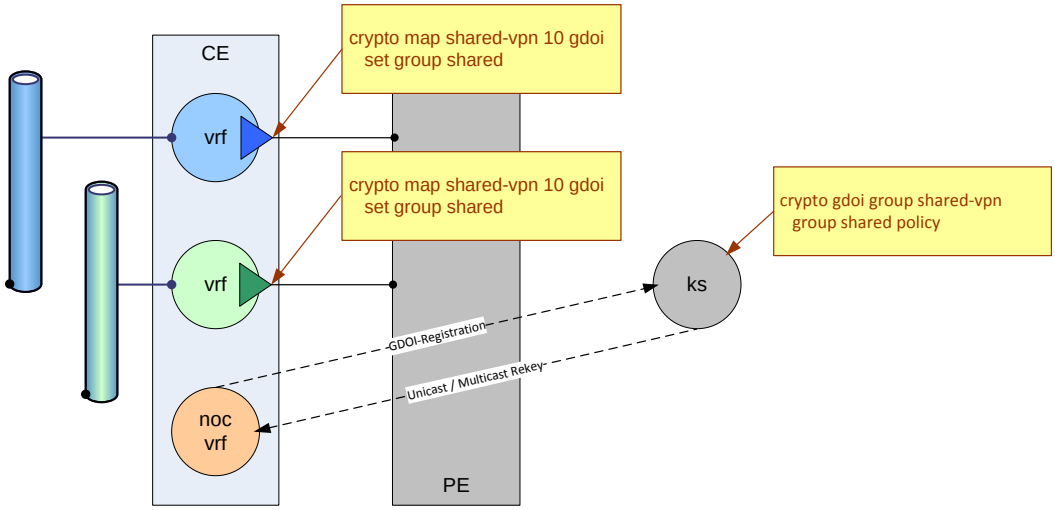
The above configuration in router behind the GM receives traffic from an interface (Gi0/0) participating in global routing to another interface (Gi0/1) with RED VRF forwarding. This way GET_GM router receives the traffic from RED VRF forwarding interface Gi0/1 and sends the traffic out encrypted in the same RED VRF forwarding interface Gi0/2.1 where crypto map is applied.

3.9.1 VRF-Aware GETVPN Deployment Scenarios

3.9.1.1 Shared GDOI groups/policies and crypto maps

The same crypto map is applied to multiple interfaces/sub-interfaces, each interface/sub-interface is in a different VRF context or the same VRF context. Key servers are accessible through a different VRF or global VRF. This can be addressed by configuring a group member with a specified registration interface. The corresponding gdoi crypto map will be applied to all the data plane interface/sub-interfaces. In such case, there is only one group member registration for all the crypto maps applied to different interfaces/sub-interfaces. After successful registration, policies are downloaded to where the crypto maps are applied. So in this case, there is one single registration and one single rekey would be received. Figure 26 illustrates this scenario:

Figure 26. Shared crypto map deployment



Use Case: Enterprise VPN that requires encryption of traffic regardless of the virtual partition. The same policy can be used on all partitions. Separation is done via routing of traffic or access control lists. For routed separation, the PE is required to use distinct VRF's per customer partition. For access control separation, the PE may use a shared VRF.

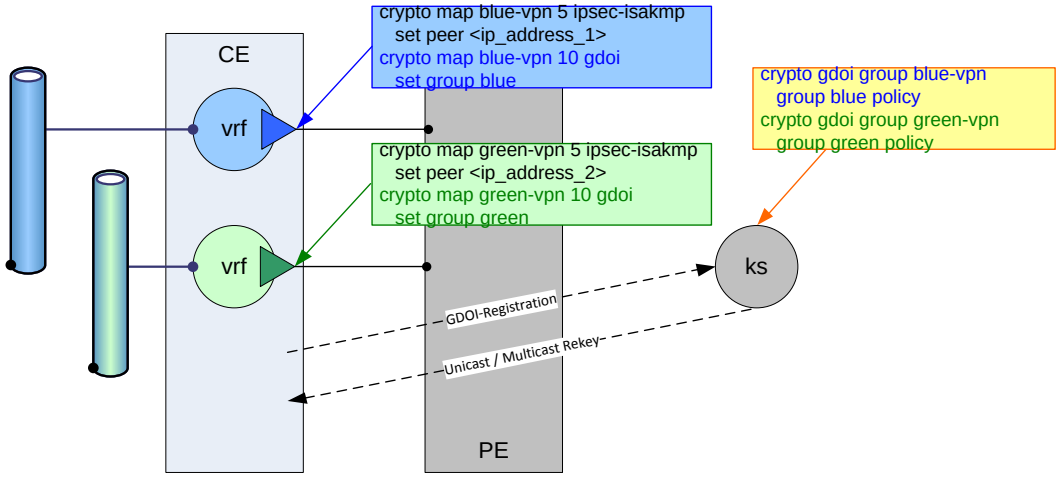
Advantage: Configuration and distribution of one group policy for all partitions.

Disadvantage: Policy distributed (permits and denies) must be universally applicable to each partition.

3.9.1.2 Different groups/policies with different crypto maps sharing same KS

In this scenario, different crypto maps assigned to separate groups are applied to different interfaces, each interface is in a different VRF context or the same VRF context. All these groups are accessing the same key server and this key server is accessible through a different VRF or global VRF table. This is addressed by having individual group members configured with the same KS and same registration interface/sub-interface. So for every group there is one single registration and also one rekey received. All these group members would be using the same IKE SA which is established between the CE and the Key server. Figure 27 illustrates this scenario.

Figure 27. Different crypto map sharing the same KS deployment



Use Case: Enterprise VPN that requires selective encryption of traffic based on the virtual partition. Most likely, unique policies will be applied. Separation may be maintained by crypto selectors with shared routing infrastructure. Routing separation may be maintained by discrete VRF's on the PE. Management control plane accessible from the enterprise.

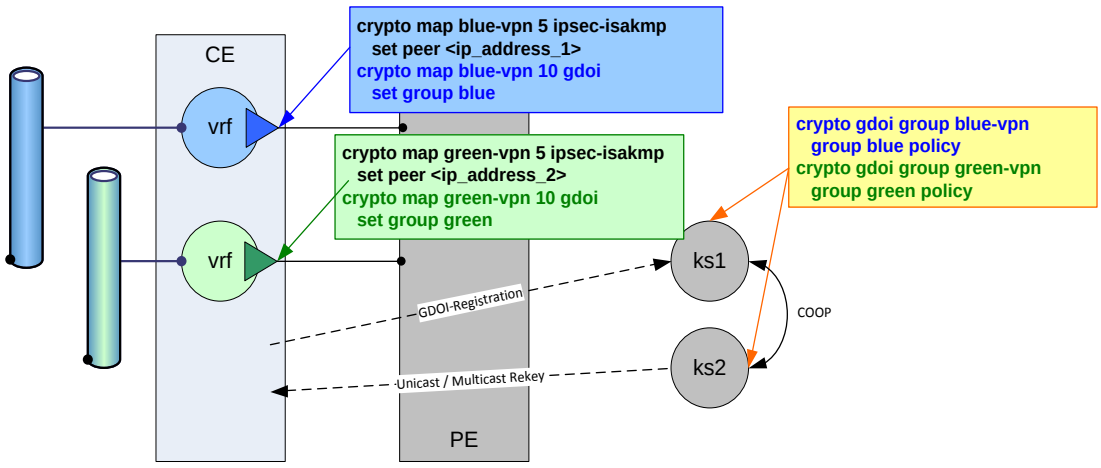
Advantage: Configuration and distribution of discrete group policy for all partitions. Application of point-to-point IPSec SA's is properly applied to two interfaces. CE identity is shared for all groups.

Disadvantage: Policy and credentials must be configured on the KS for each group.

3.9.1.3 Different groups/policies with different crypto maps on different KSs

In this deployment, different Group Member crypto maps are applied to different interfaces, where each interface is in a different VRF context or same VRF context. All these groups are accessing different Key Server and these Key Servers are accessible through different VRFs or global VRF table. This is addressed by having each group members configured with different KS and same or different registration interfaces. So for every group there is one single registration and also one rekey received. Since every registration is with a different Key server, there will be a different IKE SA established for every registration. Figure 28 illustrates this scenario:

Figure 28. Different crypto maps sharing the different KS deployment



Use Case: Enterprise VPN that requires selective encryption of traffic based on the virtual partition. Most likely, unique policies will be applied. Separation may be maintained by crypto selectors with shared routing infrastructure. Routing separation may be maintained by discrete VRF's on the PE. Management control plane accessible from the enterprise.

Advantage: Configuration and distribution of discrete group policy for all partitions. Application of point-to-point IPSec SA's is properly applied to two interfaces. CE identity shared for all groups.

Disadvantage: Policy and credentials must be configured on the KS for each group.

3.9.2 VRF-Aware Software Infrastructure (VASI) Routing

The ASR1000 platform allows the use of crypto maps on VRF-Aware Software Infrastructure (VASI) interfaces. The VASI interfaces bind two VRFs together using back-to-back internal interfaces between an 'Inside VRF' (iVRF) and a 'Front-door VRF' (fVRF). The VASI are often referred to as VASI-Left and VASI-Right respective to the iVRF and fVRF.

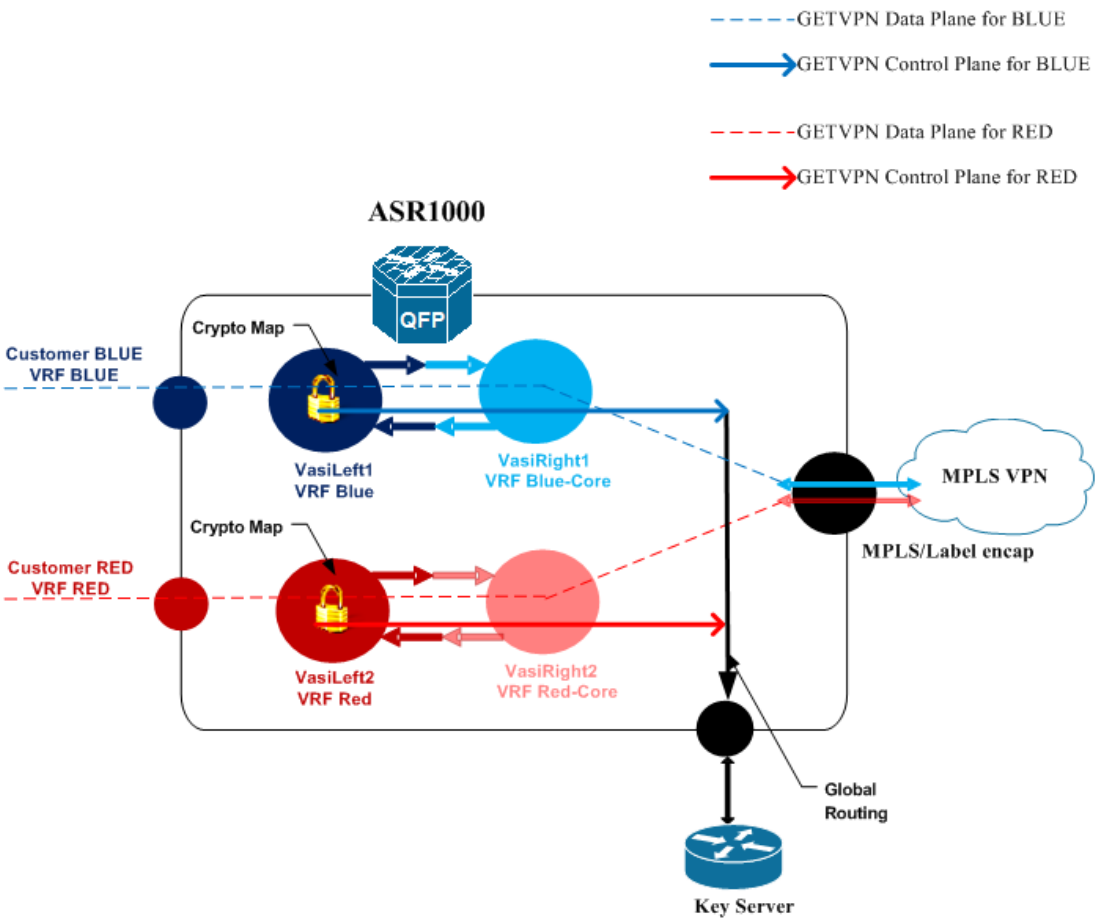
The iVRF connects to the client's private LAN environment and routes unencrypted traffic into the VASI-Left in the iVRF where a GETVPN crypto map is applied. The paired VASI-Right receives the GETVPN encrypted packet and implements a second route lookup in the fVRF. The fVRF routes the GET encrypted packet to the WAN infrastructure.

The packets routed through a shared fVRF must have unique IP address ranges for each iVRF such that return packets can be directed to the proper iVRF. GET encrypted packets arriving at an fVRF are routed to the VASI-Right interface that is bound to the VASI-Left of the appropriate iVRF. The GETVPN crypto map associated with the VASI-Left interface will decrypt the GET packet and route the packet in the iVRF associated with the client's private LAN infrastructure.

The use of VASI interfaces in the ASR allows the operator to collapse the encryption and virtual routing infrastructure into a single platform. The WAN routing methods on the fVRF are decoupled from the encryption methods on the iVRF; therefore, the choice of WAN routing methods is independent of the crypto restrictions for VRF-lite. Figure 29 illustrates the use of VASI with VRF aware GDOI.

Note: Only GDOI crypto maps are supported on VASI interfaces

Figure 29. VASI Based VRF aware GDOI on ASR1000



The following is a configuration snippet for VASI based VRF aware GDOI/GKM:

```
interface vasileft1
 ip address 10.10.10.112.255.255.255.255
 vrf forwarding VRF-Blue
 crypto map GETVPN-MAP
interface vasiright1
 ip address 10.10.20.113 255.255.255.255
 vrf forwarding VRF-Blue-Core
crypto isakmp policy 10
 encryption AES
 group 2
crypto gdoi group Blue
 identity number 1234
 server address ipv4 10.10.12.34
crypto map GETVPN-MAP local-address GigabitEthernet1/1
crypto map GETVPN-MAP 10 gdoi
 set group Blue
```

3.10 GETVPN Support for IPv6 in the Data Plane

Currently GETVPN supports GDOI protection for IPv6 traffic in the dataplane (GM-to-GM). The control plane communication (KS-to-KS and KS-to-GM) continues to use IPv4. The GM needs to be a dual-stack device in order to support ipv6 for data-plane and ipv4 for control-plane.

GETVPN with IPv6 is supported on ISRG2, ISR4k and ASR1k platforms. IPv6 support on ISRG2 platforms is enabled from 15.2(3)T release and is supported on IOS-XE based platforms from XE3.9S release. Dual stack IPv4/IPv6 support on the crypto interface of the GM is enabled in 15.2(3)T on ISR-G2 and XE3.10.3 on IOS-XE platforms.

The communication between GM and KS includes GDOI registration and rekey, and this is purely with IPv4. That means:

- On a KS, although it is configured with a IPv6 GDOI policy, registration process will use IPv4 IKE SA.
- On a GM, registration and rekey are all using IPv4 process, and “Key encrypting key (KEK)” is still an IPv4 IKE SA.

After registration or rekey, the policies downloaded from key server are IPv6 policies. The traffic encrypting keys (TEK) downloaded are IPv6 keys. The corresponding IPv6 IPSec SAs will be installed.

Note: Mixed mode under the same group is not supported. In other words, configuring both IPv4 and IPv6 policies under the same group is not supported. Configuring either IPv4 OR IPv6 policies for a single group is supported.

By configuring the crypto gdoi group ipv6 command, it is assumed that the data plane is in IPv6 and only “match address ipv6” will be allowed on the KS group configuration. Once a group is configured as “ipv6”, the configuration of the match address using IPv4 ACL is not allowed anymore. A warning is displayed and the configuration will not be accepted. Changing a GDOI group from IPv4 to IPv6 or vice-versa removes the policy and GM database corresponding to that group, and recreates them.

3.10.1 Verifying GETVPN support for IPv6

All devices in the same group (including the KS, cooperative KSs, and GMs) must support the GETVPNGETVPN for IPv6 in the Data Plane feature before the group's KS can enable the feature. The feature provides the following command that we use on the KS (or primary KS) to check whether all devices in the network are running versions that support it.

```
Router# show crypto gdoi feature ipv6-crypto-path
```

When this command is executed on the KS (or primary KS), the version of the software on each KS and GM is displayed. The output also displays whether all devices support IPv6 encryption and decryption (and thus can be added to an IPv6 group).

When executed on a GM, the command displays information only for that GM.

3.10.2 Configuring IPv6 group on the KS

Current implementation requires separate GDOI groups for IPv6 and IPv4. There is no support for mixed mode, which means that a group cannot be configured with both IPv6 and IPv4 policies.

The following configuration is for an IPv6 GDOI group on KS:

```
! IKE Phase 1 configuration remains unchanged
crypto isakmp policy 10
 encr aes
 hash sha256
 lifetime 86400
 authentication pre-share
 group 14
crypto isakmp key Cisco address 172.19.1.2
crypto isakmp key Cisco address 172.20.2.2

! IPSec transform-set and profile remains unchanged
crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
crypto ipsec profile IPSEC_PROF_GETV6
 set transform-set AES_SHA

! IPv6 policy ACL
ipv6 access-list ACL_GETV6_ANY3
 deny icmp fe80::/10 any
 deny icmp any fe80::/10
 permit ipv6 any any

! "ipv6" keyword to create a GDOI-ipv6 group
crypto gdoi group ipv6 GETV6
 identity number 1111
 server local
  rekey retransmit 10 number 2
```

```
  rekey authentication mypubkey rsa GETKEY
  rekey transport unicast
  address ipv4 172.16.4.2

! "ipv6" keyword to specify an IPv6 policy ACL
sa ipsec 1
   profile IPSEC_PROF_GETV6
   match address ipv6 ACL_GETV6_ANY3
```

3.10.3 Configuring IPv6 group on the GM

On the GM, separate crypto maps are needed for IPv6 and IPv4 traffic. IPsec crypto maps do not support mixed mode. The control plane uses IPv4 to register to the KS, download IPv6 policies, download IPv6 TEKs and to receive rekeys. When a GM downloads the policy from the KS, it confirms that the policy defines IPv6 traffic. If not, the policy is rejected, and the GM attempts to register to the next KS in its list.
The feature is supported only on those platforms that support IPv6 crypto maps. For other platforms, the applicable configuration commands are disabled.
The following configuration is for an IPv6 GDOI on GM:

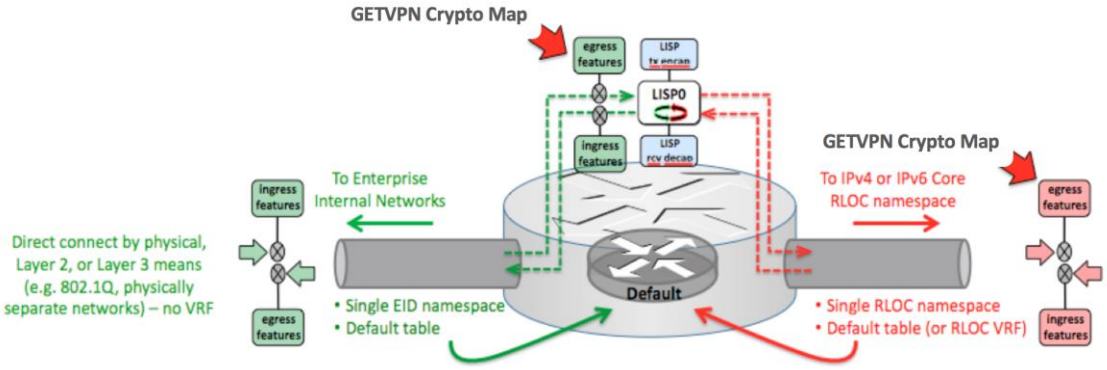
```
! IKE Phase 1 configuration remains unchanged
crypto isakmp policy 10
 encr aes
 hash sha256
 lifetime 86400
 authentication pre-share
 group 14
crypto isakmp key Cisco address 172.16.4.2
crypto isakmp key Cisco address 172.18.5.2

! "ipv6" keyword to create a GDOI-ipv6 group
crypto gdoi group ipv6 GETV6
  identity number 1111
server address ipv4  172.16.4.2
server address ipv4  172.18.5.2

! Configure IPv6 gdoi crypto map
crypto map ipv6 CM_GET_V6 10 gdoi
 set group GET_GRP_V6
```

```
 match address GM_ACL_V6


! IPv6 GDOI cryptomap on interfaces with dual-stack
int GigabitEthernet0/0
  ip address 172.19.1.2 255.255.0.0
  ipv6 address 2001:DB8:FFFF::2/64
  ipv6 crypto map CM_GET_V6
```

3.11 GETVPN Support for LISP

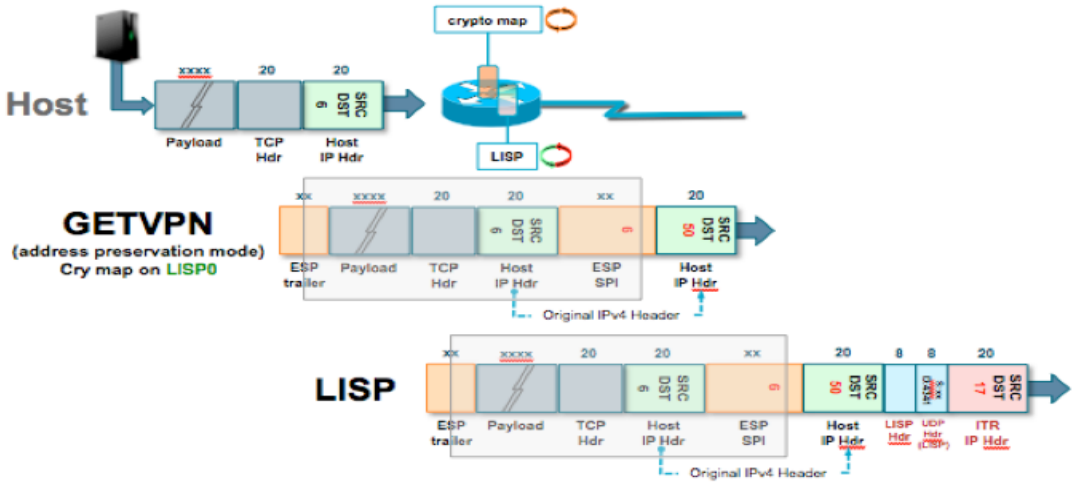
The inherent properties of LISP give it support for multi-homing, virtualization, and host/VM mobility for both IPv4 and IPv6 address families make it an ideal architecture for creating highly efficient, AF-agnostic, Virtual Private Networks (VPNs). Existing IOS encryption support provided by the IPsec and GETVPN features can be used directly (in a “bolt-on” manner) with LISP to build encrypted VPNs.Because LISP separates locators and endpoint identifiers, encryption can be added using IPsec or GETVPN by applying the crypto map to either the EID side (LISP0 virtual interface), or to the locator side (RLOC interface(s)), as illustrated in Figure 30.

Figure 30. Crypto Map Application Points Available to “bolt-on” GETVPN with LISP



Depending on where the crypto map is applied (as per Figure 1), the resulting configuration details change, as does the resultant packet handling and encrypted packet format. With GETVPN+LISP solution, crypto map is applied to the LISP0 virtual interface. This is the most common architecture and provides the most flexibility for applying unique security policies within the resultant VPN environment. When applied to **LISP0**, GETVPN encryption occurs first, followed by LISP encapsulation. The packet construction process is illustrated in Figure 31 below.

Figure 31. LISP with GETVPN applied to LISP0 results in GETVPN and then LISP



To configure GETVPN crypto map with LISP interfaces,

```
interface LISP0
!
interface LISP0.1
  ip mtu 1456
  ipv6 mtu 1436
  ipv6 crypto map MAP-V6-0001
  crypto map MAP-V4-0001
interface LISP0.2
  ip mtu 1456
  ipv6 mtu 1436
  ipv6 crypto map MAP-V6-0002
  crypto map MAP-V4-0002
interface LISP0.3
  ip mtu 1456
  ipv6 mtu 1436
  ipv6 crypto map MAP-V6-0003
  crypto map MAP-V4-0003
```

The LISP process automatically creates each LISP0.x virtual interfaces once an IID is configured. LISP0 (default table) should not have a crypto map and thus incurs no encryption. Only the LISP0.x interfaces associated with the Departmental VPNs are encrypted – each with its own policy, and on a per address-family basis as well.

For more information on LISP deployment details with GETVPN, refer to the resources below

https://www.cisco.com/c/dam/en/us/td/docs/solutions/Enterprise/Data_Center/DCI/5-0/GETVPN.pdf

http://lisp.cisco.com/docs/GETVPN_LISP_Deployment_1.0.pdf

http://lisp.cisco.com/lisp_tech.html#CG

3.12 GETVPN GDOI Bypass

The GETVPN GDOI Bypass feature allows enabling or disabling the GDOI bypass crypto policy and control traffic exceptions by explicitly configuring the GM local access control list (ACL).

When a GDOI crypto-map is applied to an interface, a default crypto policy is installed to allow GDOI protocol traffic (UDP port 848 sourced from any IP address and destined to any IP address) to pass through in cleartext. The GETVPN GDOI Bypass feature improves the security of the default GDOI bypass crypto-policy when handling UDP848 traffic in the following ways:

- Enable and disable the default GDOI bypass crypto policy through the a new CLI on the GM **client bypass-policy**
- Hardening the GM's default GDOI bypass crypto policy when enabled

The GDOI Bypass policy will be installed only on GETVPN-protected interfaces. By default, GDOI client bypass-policy is enabled to ensure backward compatibility. When GDOI bypass is enabled, the following will be seen on the GM:

```
GM# show crypto gdoi gm acl
Group Name: GETVPN
 ACL Downloaded From KS 10.0.0.2:
   access-list   deny eigrp any any
   access-list   permit ip any any
 ACL Configured Locally:
 ACL of default GDOI bypass policy:
   Ethernet1/0: deny udp host 10.0.0.9 eq 848 any eq 848 vrf RED*
```

For more information on this feature, refer to the GDOI Bypass in GETVPN Configuration Guide..

3.13 GETVPN Routing Awareness

The GETVPN Routing Awareness feature prevents routing black hole by tracking the GETVPN GM crypto state and by applying the tracking information to perform bidirectional conditional route filtering on the GM. The GETVPN Routing Awareness feature prevents routing black hole by tracking the GETVPN GM crypto state and by applying the tracking information to perform bidirectional conditional route filtering on the GM. The routing plane and crypto plane for GETVPN must be synchronized to avoid blackholes. Before a GM successfully registers to a KS, no security policies or keys are installed on the GM; however, the GM may still advertise the routes of its protected network to other GMs. The command "client status active-sa track" was introduced to enable GETVPN routing awareness. The feature is available since Cisco IOS-XE Release 3.13S.

For more information on routing awareness and configuration examples, refer to GETVPN Routing Awareness in GETVPN Configuration Guide.

3.14 IPsec Inline Tagging for Cisco TrustSec on GETVPN

Cisco TrustSec (CTS) architecture secures networks by establishing domains of trusted network devices. Once a network device authenticates with the network, the communication on the links between devices in the cloud is secured with a combination of encryption, message integrity checks, and replay protection mechanisms.

CTS uses the user and device identification information acquired during the authentication phase to classify packets as they enter the network. CTS maintains classification of each packet or frame by tagging it with a security group tag (SGT) on ingress to the network so that it can be identified for applying security and other policy criteria along the data path. The tags allow network intermediaries such as switches and firewalls to enforce access control policy based on the classification.

The GETVPN Support of IPsec Inline Tagging for Cisco TrustSec feature uses GETVPN inline tagging to carry the SGT information across the private WAN.

For information on restrictions, more conceptual information, configuration, and verification examples, go to GETVPN Support of IPsec Inline Tagging for Cisco TrustSec document.

When a KS receives a security association (SA) registration request from a group member (GM) or receives a connection establishment request from a cooperative KS, it checks whether any group SA has SGT inline tagging enabled. If so, all GMs and cooperative KSs must register using GETVPN software version 1.0.5 or higher to be accepted. Otherwise, the registration request or establishment request is rejected, and the KS generates a syslog message to notify the network administrator.

The following example shows how to use the GETVPN software versioning command on the KS (or primary KS) to check whether all the devices in each group support IPsec inline tagging for Cisco TrustSec:

```
Device# show crypto gdoi feature cts-sgt

Group Name: GETVPN
    Key Server ID       Version   Feature Supported
        10.0.5.2         1.0.5          Yes
        10.0.6.2         1.0.5          Yes
        10.0.7.2         1.0.3          No
        10.0.8.2         1.0.2          No

    Group Member ID     Version   Feature Supported
        10.0.1.2         1.0.2          No
        10.0.2.5         1.0.3          No
        10.0.3.1         1.0.5          Yes
        10.0.3.2         1.0.5          Yes
```

You can also enter the above command on a GM (which displays the information for the GM but not for the KS or other GMs).

The following example shows how to enter the command on the KS (or primary KS) find only those devices in the GETVPN network that do not support IPsec inline tagging for Cisco TrustSec:

```
Device# show crypto gdoi feature cts-sgt | include No
```

```
        10.0.7.2         1.0.3          No
        10.0.8.2         1.0.2          No
        10.0.1.2         1.0.2          No
        10.0.2.5         1.0.3          No
```

The following configuration snippet configures IPsec inline tagging on a KS:

```
ip access-list extended ACL-SGT
 permit ip any any
exit
 crypto gdoi group GET-SGT
  identity number 1
  server local
    sa ipsec 1
     tag cts sgt
     profile gdoi-p2
     match address ipv4 ACL-SGT
     replay time window-size 100
```

The following configuration snippet configures IPsec inline tagging on a GM. In this configuration, there are two groups: A group with GMs that are upgraded to GETVPN version 1.0.5 or higher (and therefore supports CTS SGT inline tagging) and a group with GMs that are not yet upgraded. The upgraded GMs will register to group number 1111 (a lower crypto map sequence number) and with group number 2222 (a higher crypto-map sequence number). Non-upgraded GMs will register only to group number 2222.

In this example SGT tagging for traffic is configured between two sites. The permit ip commands add access control entries (ACEs) to the access control list (ACL) that permit communication between the two sites:

```
ip access-list extended ACL_NET_AB
 permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
 permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
 exit
ip access-list extended ACL_ALL
 permit ip any any
 exit
crypto gdoi group GET1
 identity number 1111
 server local
  rekey authentication mypubkey rsa mykey
  rekey transport unicast
  sa ipsec 1
   tag cts sgt
   profile gdoi-p2
   match address ipv4 ACL_NET_AB
   replay time window-size 100
   exit
  exit
```

```
exit
crypto gdoi group GET2
 crypto gdoi group GET2
  server local
   rekey authentication mypubkey rsa mykey
   rekey transport unicast
   sa ipsec 1
    profile gdoi-p2
    match address ipv4 ACL_ALL
    replay time window-size 100
```

3.15 GETVPN Software versioning

A versioning system has been implemented in GETVPN software to simplify feature compatibility issues. GDOI SW version has three parts – Major_version, Minor_version, and Mini_version in the following format: Major_version.Minor_version.Mini_version

- Major_version defines compatibility for all GETVPN devices.
- Minor_version defines compatibility for KS-to-KS (cooperative key server) associations and for GM-to-GM interoperability.
- Mini_version tracks feature changes that have no compatibility impact.

Important notes about version compatibility:

1. All GMs and KS must have the same “major” version
2. All GMs must have the same “minor” version
3. All KSs must have the same “minor” version
4. Version 1.0.1 is the base-version for releases up to IOS version 15.1(4)M and IOS XE version 15.2(4)S or XE 3.7
5. A base-version KS (1.0.1) does not send GM version info to other COOP-KS. Thus, all GMs registered to base-version KS will be shown up as version “unknown” on the other COOP-KS

Following is an example matrix that shows KS/GM compatibility based on the version. If the KS and GM are not compatible, the GM registration will not succeed.

	GM Version			
KS Version		Base (1.0.1)	1.0.2	1.1.0
	Base (1.0.1)	Yes	Yes	Yes
	1.0.2	Yes	Yes	Yes
	1.1.0	Yes	Yes	Yes

Following is an example matrix that shows KS/KS compatibility based on the version. If the KSs are not compatible, coop election will not succeed.

	KS2 Version

KS1 Version		Base (1.0.1)	1.0.2	1.1.0
	Base (1.0.1)	Yes	Yes	No*
	1.0.2	Yes	Yes	No*
	1.1.0	No*	No*	Yes

* Warning syslog will be printed about version mismatch

Commands to check GDOI version and feature compatibility:

1. GDOI version check of all COOP-KS

```
KS701# sho cry gdoi ks coop
Crypto Gdoi Group Name :GET
        Group handle: 2147483650, Local Key Server handle: 2147483650
        Local Address: 10.0.8.1
        Local Priority: 10
        Local KS Role: Primary   , Local KS Status: Alive
        Local KS version: 1.0.3
        …….
        Peer Sessions:
        Session 1:
                Server handle: 2147483651
                Peer Address: 10.0.9.1
                Peer Version: 1.0.3
```

2. GDOI version check of all GMs from the primary KS

```
KS701#sho cry gdoi ks members
Group Member Information :
Number of rekeys sent for group GET : 4
Group Member ID    : 5.0.0.2     GM Version: 1.0.2
 Group ID          : 3333
 Group Name        : GET
 Key Server ID     : 10.0.8.1
 ……..
Group Member ID    : 9.0.0.2     GM Version: 1.0.1
 Group ID          : 3333
 Group Name        : GET
 Key Server ID     : 10.0.9.1
 ……
```

3. The following command illustrates feature compatibility with the use of versions

```
show crypto gdoi feature [group <grp_name>] feature <feature_name>
```

```
KS701# show crypto gdoi feature ?
  gdoi-mib        GDOI Management Information Base support
  gm-removal      RFC defined delete message to remove GM's policy
  policy-replace  Replace current policy in the rekey message


KS701# show crypto gdoi feature gm-removal
Group Name: GET
    Key Server ID          Version      Feature Supported
        10.0.8.1           1.0.3               Yes
        10.0.9.1           1.0.3               Yes

    Group Member ID        Version      Feature Supported
        5.0.0.2            1.0.3               Yes
        9.0.0.2            1.0.1               No
```

The following table provides information about the different GDOI versions in IOS and IOS-XE. The IOS/IOS-XE versions listed in the table are the first versions that will have the corresponding GDOI version.

Table 3. Mapping between GDOI versions and IOS /IOS-XE releases

GDOI Version	IOS Release (M&T Train)	IOS-XE Release (3S/S/XE16 Train)
Unknown	15.2(1)T	N/A
1.0.1	15.2(1)T	15.3(1)S/XE 3.8S
1.0.2	15.2(1)T – 15.2(2)T	15.3(1)S / XE3.8S – 15.3(1)S2/XE3.8.2S
1.0.3	15.2(3)T	15.3(1)S/XE 3.8S
1.0.4	15.2(4)M-	XE 3.9S
1.0.5	N/A	15.3(2)S/XE3.9S
1.0.6	15.3(2)T	XE 3.9S
1.0.7	N/A	15.3(3)S / XE3.10S
1.0.8	15.3(3)M	15.3(3)S/XE XE 3.10S, XE 3,10.1S
1.0.9	15.4(2)T	15.4(2)S/XE 3.12S
1.0.10	15.4(3)M	15.4(3)S/XE 3.13S
1.0.11	N/A	N/A
1.0.12	15.5(1)T	15.5(1)S/XE 3.14S (GM only)
1.0.13	15.5(1)T	15.5(1)S/XE 3.14S (KS only)
1.0.14	15.5(2)T	15.5(2)S/XE3.15S
1.0.15	15.5(1)T	15.5(1)S/XE 3.14S
1.0.16	15.5(3)T	15.5(3)S/XE3.16S
1.0.17	N/A	Everest 16.5
1.0.18	15.5(3)T	15.5(3)S/XE 3.16S, Everest 16.5
1.0.19	N/A	Fuji 16.7.1

For detailed information about the features supported for these GDOI versions, please refer to the following documents:

Cisco Group Encrypted Transport VPN Configuration Guide, Cisco IOS Release 15M&T

Cisco Group Encrypted Transport VPN Configuration Guide, Cisco IOS XE Fuji 16.8.x

3.16 GM Removal and Policy Replacement

This feature facilitates easy removal of unwanted GMs from the GETVPN network; it also provides a rekey method to install new SAs and remove obsolete SAs. This feature is supported from 15.2(1)T in IOS and 15.3(1)S or IOS-XE 3.8 in IOS XE.

3.16.1 GM Removal

Prior to the GM removal feature, to delete an unwanted GM from a group it was required to complete the following steps:

1. Revoke the Phase-1 credentials (for example, the preshared key or one or more PKI certificates).
2. Clear the TEK and KEK database on all Key Servers.
3. Clear the TEK and KEK database on each GM individually to force each GM to re-register.

The third step is time-consuming when there are many Group Members. In addition, clearing an entire group manually would cause a network disruption. This feature automates this process by introducing a command that is entered on the primary KS to create a new set of TEK and KEK keys and force each GM to re-register.

Below are the steps to remove an unwanted GM from the group:

1. Revoke IKE Phase-1 credentials of unwanted GM from all the KSs
2. Execute the following command on the primary KS. This will send a delete message to all secondary-KS and GMs

clear cry gdoi [group <grp-name>] ks members [now]

3. All KSs delete old TEK/KEK/GM database and install new TEK/KEK
4. All GMs cleanup TEK/KEKs individually and re-register
5. KS will only accept GM registration based on new IKE credentials, thereby blocking the unwanted GM from registering with any of the KSs in the group.

3.16.1.1 GM Removal with Transient IPsec SAs

With the use of transient IPsec SAs, we can avoid disruption in the network as traffic continues to be encrypted and decrypted using the transient IPsec SA until its lifetime expires. Following steps outline this method of GM removal process:

- Revoke the IKE phase 1 credentials of unwanted GM from all the KSs
- Issue the following clear command on the primary KS

```
Primary-KS701#clear crypto gdoi ks members
% This GM-Removal message will shorten all GMs' key lifetimes and cause them
to re-register before keys expiry.
Are you sure you want to proceed? [yes/no]: yes
Sending GM-Removal message to group GET...
```

- The GMs will generate this syslog message:

```
*Jan 28 08:37:03.103: %GDOI-4-GM_RECV_DELETE: GM received delete-msg from KS
in group GET. TEKs lifetime are reduced and re-registration will start
before SA expiry
```

- GMs delete KEK immediately and reduce old TEK lifetime using the following fomula

TEK_SLT = MIN(TEK_RLT, MAX(90s, MIN(5%(TEK_CLT), 3600s))

TEK_SLT: shortened lifetime; TEK_RLT: Remaining LIfeTime; TEK_CLT: Configured LifeTime

- All GMs except the revoked GM will re-register to get new TEK/KEK by using the conventional re-registration timer with the jitter applied.

- No network disruption is expected as traffic continues to flow by using transient IPSEC SAs

3.16.1.2 GM Removal with Immediate IPsec SA Deletion

This method of GM removal is used to force GMs to delete old TEKs and KEKs immediately (without using transient SAs) and re-register. However, this can cause a disruption to the data plane, so this method should be used only if there are important security concerns that require removal of a GM immediately. Following steps outline this method of GM removal process:

- Revoke the IKE Phase-1 credentials of unwanted GM from all the KSs
- Issue the following clear command on the primary KS

```
Primary-KS701#clear crypto gdoi ks members now
% This GM-Removal immediate message will cleanup all GMs downloaded policies
% This will cause all GMs to re-register.
Are you sure you want to proceed? [yes/no]: yes
Sending GM-Removal message to group GET…
```

- The above command triggers the GMs to do the following:
  - Delete all TEKs/KEK/policy immediately
  - GMs go to fail-open mode unless fail-close is explicitly configured
  - All GMs except the revoked GM will re-register successfully within a randomly chosen period MIN(TEK_RLT, MIN(2%(TEK_CLT), 3600s))
- The GMs will generate this syslog message:

```
*Jan 28 08:27:05.627: %GDOI-4-GM_RECV_DELETE_IMMEDIATE: GM receive REMOVAL-
NOW in group GET to cleanup downloaded policy now. Re-registration will
start in a randomly chosen period of 34 sec
```

3.16.1.3 Limitations of GM Removal

The delete message can only be sent from the primary KS. The command is rejected when executed on a secondary KS with the following message:

```
Secondary-KS702#clear crypto gdoi ks members
ERROR for group GET: can only execute this command on Primary KS
```

All the KSs and GMs should be upgraded to a supported version for this feature to work. Else, KS/GMs will ignore delete-msg and continue to use old SAs resulting in traffic drop. The following warning message is displayed in case some devices do not support this feature.

```
KS701#clear crypto gdoi ks members
WARNING for group GET: some devices cannot support GM-REMOVAL and can cause
network disruption. Please check 'show crypto gdoi feature'.
Are you sure you want to proceed ? [yes/no]: no
```

It is possible to check which devices support the feature using the following command on the primary KS:

```
Primary-KS701# show crypto gdoi feature gm-removal
Group Name: GET
    Key Server ID             Version      Feature Supported
        10.0.8.1              1.0.3               Yes
        10.0.9.1              1.0.3               Yes
    Group Member ID     Version      Feature Supported
        5.0.0.2               1.0.3               Yes
        9.0.0.2               1.0.1               No
```

3.16.2 Triggered Rekey and Policy Replacement

This feature provides a mechanism of triggering rekeys to remove old/obsolete KEK and TEK while installing new ones after a policy change. The reson for this feature is to streamline the rekey process by eliminating

inconsistent behavior in the old rekey method. Following table shows the inconsistencies with the old rekeying method:

Table 4. Inconsistencies with the old rekey method

Policy Changes	Rekey Sent Immediately?	Rekey Behavior After Policy Changes
TEK: SA lifetime	No	The old SA remains active until its lifetime expires. The new lifetime will be effective after the next scheduled rekey.
TEK: IPSEC transform set	Yes	The SA of the old transform set remains active until its lifetime expires.
TEK: IPSEC profile	Yes	The SA of the old profile remains active until its lifetime expires.
TEK: Matching ACL	Yes	Outbound packet classification immediately uses the new access control list (ACL), however the old SAs remain in the SA database.
TEK: Enable replay counter	Yes	The old SA without counter replay remains active until its lifetime expires.
TEK: Change replay counter value	No	The SA with a new replay counter is sent out in the next scheduled rekey.
TEK: Disable replay counter	Yes	The old SA with counter replay enabled remains active until its lifetime expires.
TEK: Enable TBAR	Yes	The old SA with TBAR disabled remains active until its lifetime expires.
TEK: Change TBAR window	No	The SA with a new TBAR window will be sent out in the next scheduled rekey.
TEK: Disable TBAR	Yes	The old SA with TBAR enabled remains active until its lifetime expires.
TEK: Enable receive-only	Yes	Receive-only mode is activated right after the rekey.
TEK: Disable receive-only	Yes	Receive-only mode is deactivated right after the rekey.
KEK: SA lifetime	No	The change is applied with the next rekey.
KEK: Change authentication key	Yes	The change is applied immediately.
KEK: Change crypto algorithm	Yes	The change is applied immediately.

With this feature, policy changes alone will no longer trigger an immediate rekey. When a policy change is made (and the administrator exits from global configuration mode), a syslog message appears on the primary KS indicating that the policy has changed and a rekey is needed. This feature provides a new command that can then be entered on the primary KS to send a rekey. The following example illustrates this feature:

```
Primary-KS701#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Primary-KS701(config)#crypto gdoi group GET
Primary-KS701(config-gdoi-group)#server local
```

```
Primary-KS701(gdoi-local-server)#sa ipsec 1
Primary-KS701(gdoi-sa-ipsec)#no profile gdoi-p1
Primary-KS701(gdoi-sa-ipsec)#profile gdoi-p2<= changing the ipsec profile
KS701(gdoi-sa-ipsec)#end
Primary-KS701#
*Jan 28 09:15:15.527: %SYS-5-CONFIG_I: Configured from console by console
*Jan 28 09:15:15.527: %GDOI-5-POLICY_CHANGE: GDOI group GET policy has changed.
Use 'crypto gdoi ks rekey' to send a rekey, or the changes will be send in the
next scheduled rekey


Primary-KS701#crypto gdoi ks rekey
   *Jan 28 09:17:44.363: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey
   with policy-replace for group GET from address 10.0.8.1 with seq # 2
```

Once the triggered rekey command is issued on the primary KS, GMs install new SAs and reduce the old SA lifetime using the following formula:

TEK_SLT = MIN(TEK_RLT, MAX(90s, MIN(5%(TEK_CLT), 3600s)))

TEK_SLT: TEK shortened lifetime, TEK_RLT: TEK Remaining LIfeTime, TEK_CLT: TEK Configured LIfeTime

If the GMs need to be rolled over to the new SAs immediately, we can use the “replace-now” option of the triggered rekey. This option might cause traffic to drop during the switchover as the old SAs are deleted immediately. The command is:

Primary-KS701#crypto gdoi ks rekey replace-now

Triggered Rekey restrictions:

1. Triggered rekey can be done only on the primary KS. The command is rejected when executed on the secondary KS. The following error message is displayed:

```
Secondary-KS702#cry gdoi ks rekey
```

ERROR for group GET: This command must be executed on Primary-KS

2. All GMs and KSs in the GETVPN network must support this feature for the triggered rekey to be effective. The following behavior occurs when any of the KSs/GMs do not support this feature:

- The KS will only send a triggered rekey with NO policy-replacement.
- The GMs will install the new SAs but it will NOT reduce old SA lifetime

It is possible to determine which of the KSs/GMs support this feature using the following command on the Primary KS:

```
Primary-KS701#show crypto gdoi feature policy-replace
Group Name: GET
    Key Server ID             Version      Feature Supported
        172.16.4.2            1.0.2               Yes
        172.18.5.2            1.0.2               Yes
    Group Member ID           Version      Feature Supported
        172.19.1.2            1.0.2               Yes
        172.20.2.2            1.0.1               No
```

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 GETVPNGETVPN networks into an existing customer deployment.

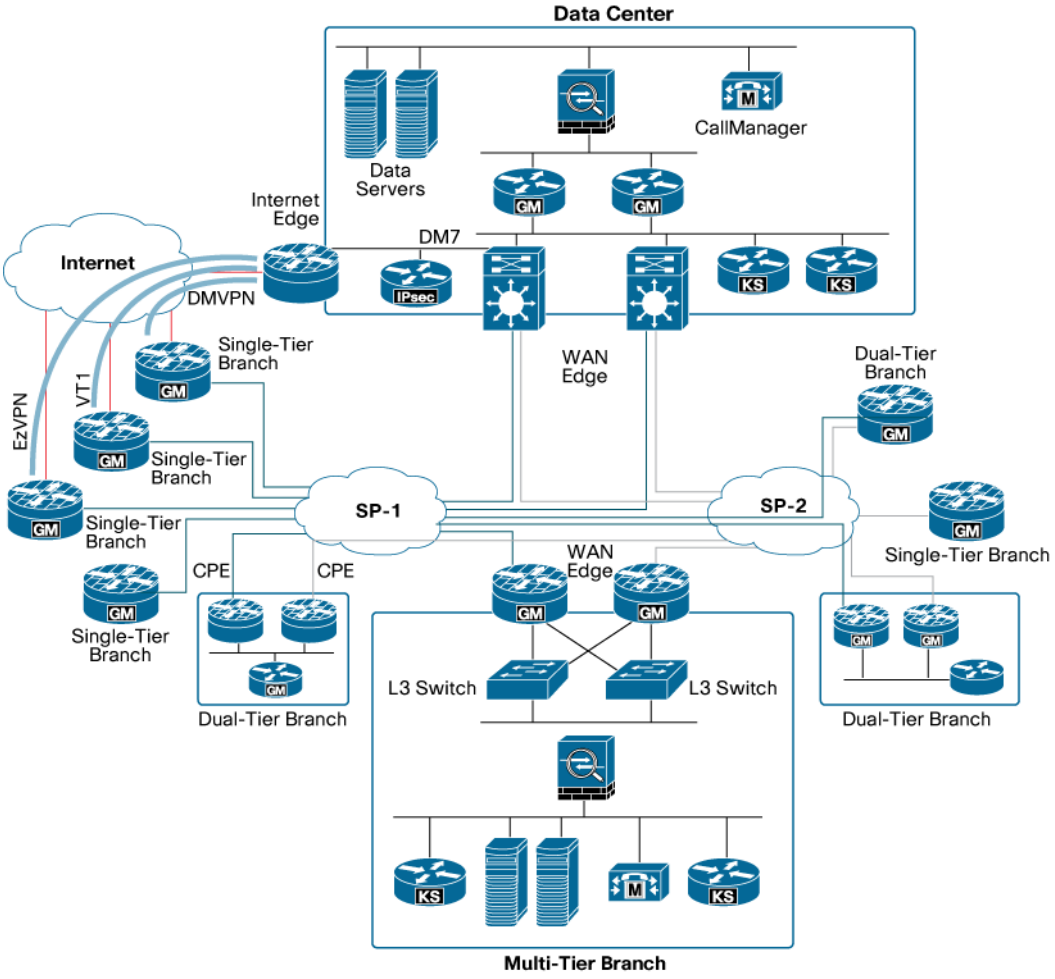
4.1.1 DC Design

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 GETVPN can be deployed over it. As part of design considerations, the effect of GETVPN on the DC components is explained. We also consider the placement of GETVPN 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 32. 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 GETVPN 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 FlexVPN 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 GETVPN 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); GETVPN 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. GETVPN 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.

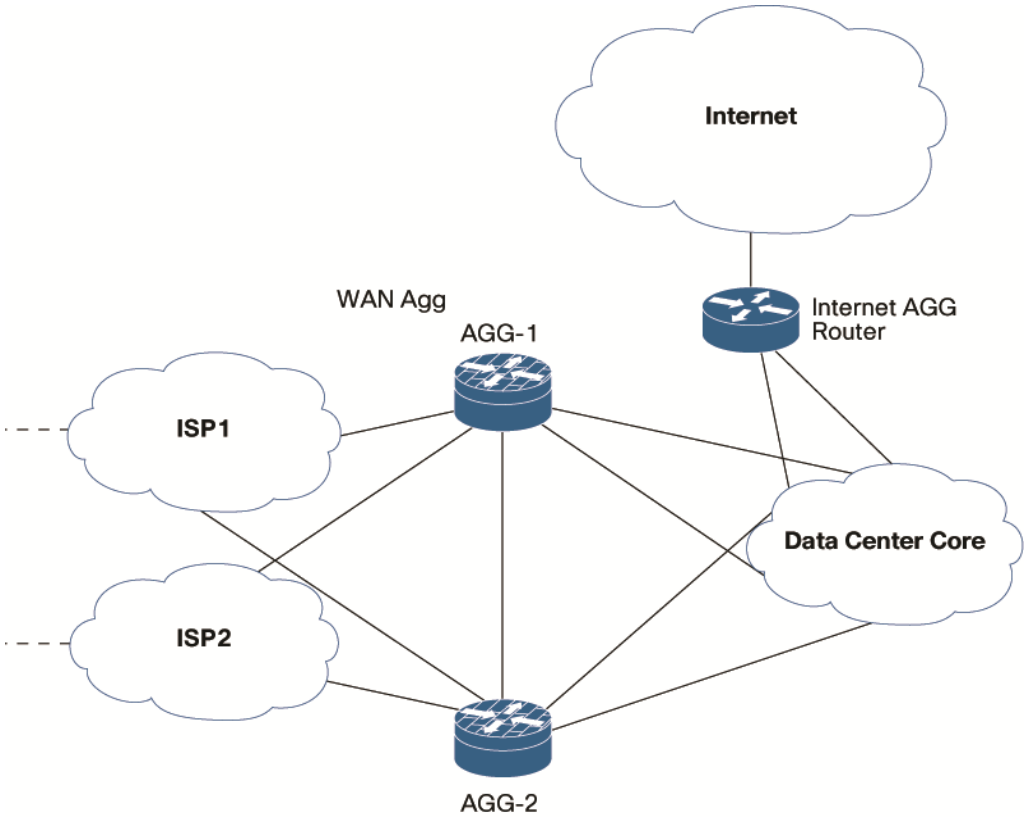
Note: If the GETVPN network is terminated on the WAN termination devices and a KS is also deployed in the branch (as shown in the preceding topology), KS policies must be configured to permit control plane traffic to and from the KS to go through the GM without IPsec encryption.

4.2 DC Design

This section examines GETVPN 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 33 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 33. DC before GETVPN Deployment



4.2.1 GETVPN DC Design Considerations

To deploy GETVPN 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**: GETVPN 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

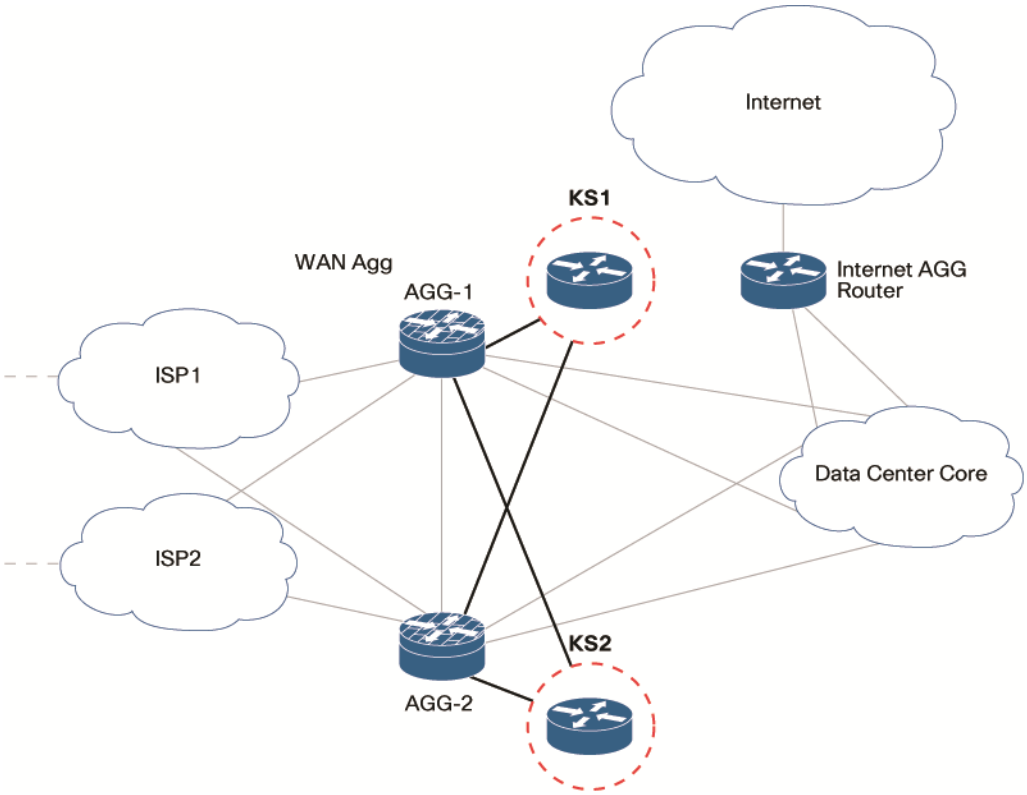
With redundant KSs in the system, KSs can be placed in different locations/sites or they can be put together on a single site. See 3.7.3.1 “Selecting the Number of COOP KSs” and 3.7.3.2 “Setting KS Priority” for recommendations for how many KS to consider and choosing KS priorities. In the case of a geographically disperse GETVPN 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 34 shows one of the ways in which the KS can be placed in the DC.

Figure 34. 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.

Placement of KS behind GM

In deployments where KS is placed behind a GM, when unicast rekey is used, the KS policy should have an ACL entry to avoid encryption of the GDOI and COOP control plane traffic. Following is an example of the ACL entry:

deny udp any eq 848 any eq 848

This policy should be applied to any GM where the GDOI and COOP traffic must transit. If there are only a small number of GMs, this entry can be added as a local policy on the GMs to avoid encrypting the control plane. On larger deployments, it would be easier to include the deny in the KS policy that is downloaded to all GMs.

4.2.1.1.2 KS Redundancy

As shown in Figure 34, 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 or ASA 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 172.16.1.1 255.255.0.0
!
```

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
!
```

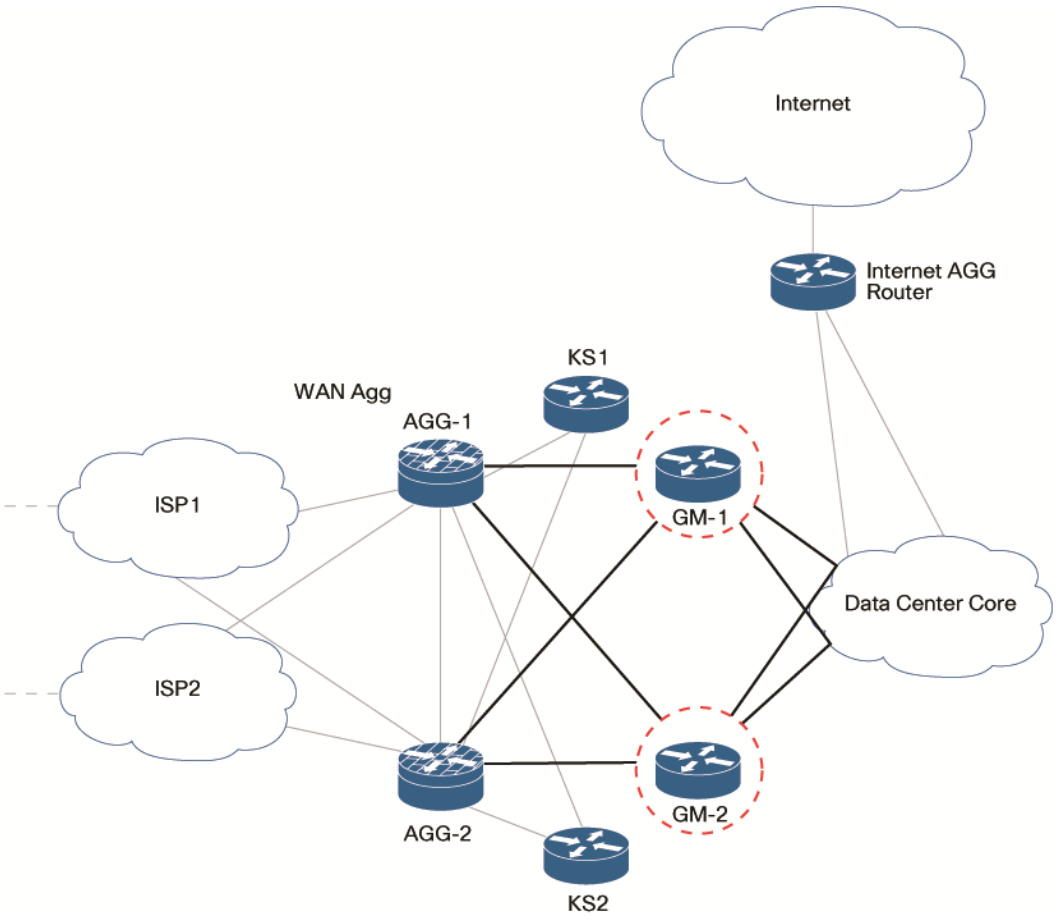
4.2.1.2 GM Design

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 33 shows GM router placement.

Figure 35. 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 Figure 35, each GM has dual links to the WAN aggregation devices and to the DC core. As an example, in the DC shown in Figure 32, GM1 has two routes to either KS.

```
GM1#sh ip route 172.16.4.2                    <-- KS1 = 172.16.4.2
Routing entry for 172.16.4.2/32
 Known via "ospf 10", distance 110, metric 14, type intra area
 Last update from 172.16.1.1 on GigabitEthernet0/1, 1w1d ago
 Routing Descriptor Blocks:
 * 172.16.1.1, from 172.16.4.2, 1w1d ago, via GigabitEthernet0/3
Route metric is 14, traffic share count is 1
<-- G0/3 is link to AGG-2
 172.16.1.1, from 172.16.4.2, 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 35 shows how two GMs are deployed.

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

Consider an example in which network 172.18.0.0/16 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 172.18.0.0
Routing entry for 172.18.0.0/16
 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 172.18.1.1, 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 172.19.1.1, 6d21h ago, via GigabitEthernet0/1
Route metric is 10, traffic share count is 1
Route tag 500
<-- G0/1 is link to GM-1
```

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 172.18.0.0 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.

Note: When local address is configured for a crypto map with gdoi group, the local address interface and the cryptomap interface should be in the same VRF.

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.

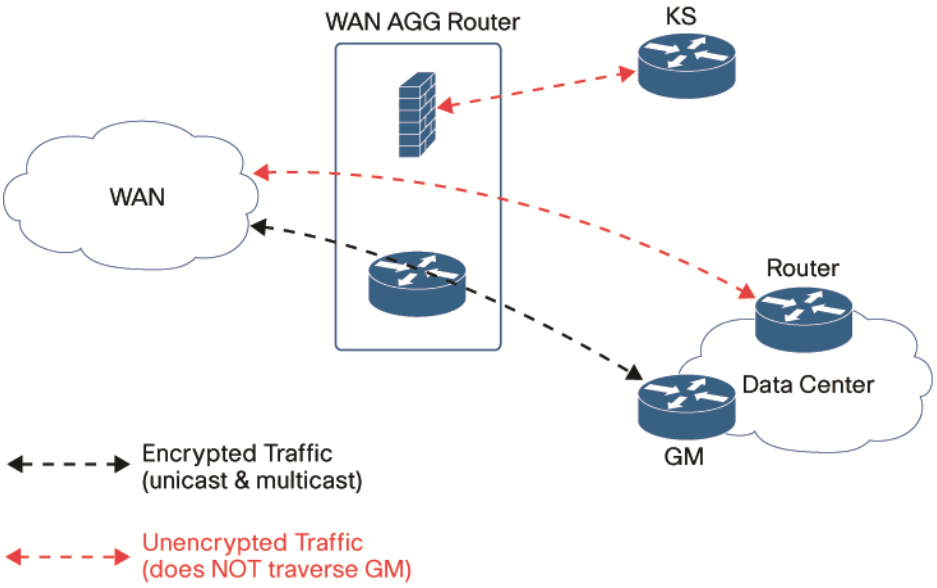
Certain firewalls, such as Cisco ASA or FWSM modules (for Catalyst 6500) might not be able to handle encrypted multicast traffic, i.e., the firewall cannot forward Encapsulating Security Payload (ESP) encapsulated multicast traffic. In such scenarios, it is recommended to place the GMs outside of the 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 36 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 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 34.

Figure 36. 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.

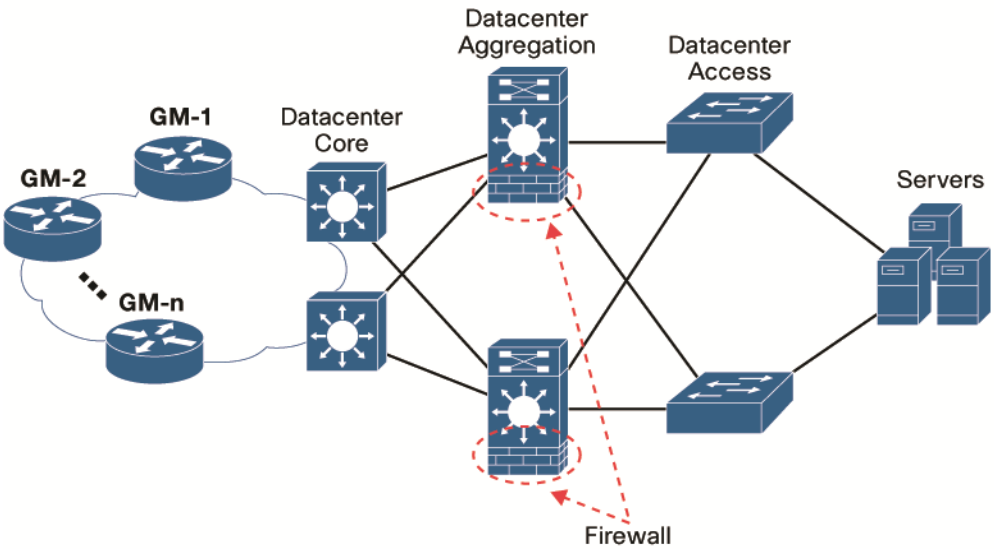
```
!
access-list 100 permit pim any any
access-list 100 permit esp any any
access-list 100 permit udp any any eq 848
access-list 100 permit ospf any any
access-list 100 deny   ip any any
!
! SVI - Vlan20 terminates GM connections on 6500
!
interface Vlan20
```

```
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 37 shows a typical DC with firewall services as part of the DC Aggregation layer.

Figure 37. Additional Firewall Services Deployed in the DC



4.2.1.3 QoS

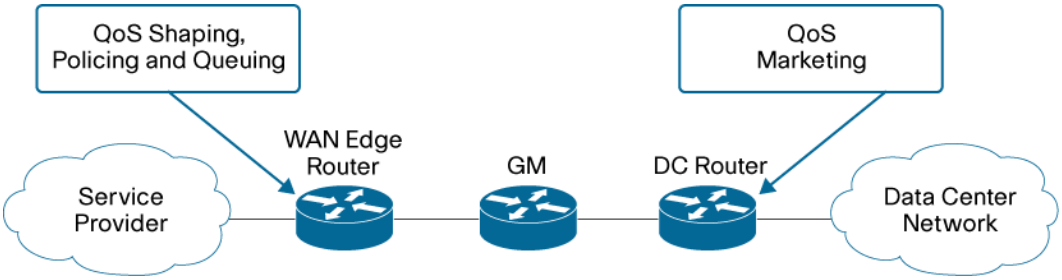
In a DC design, it is recommended not to deploy QoS functionality on the GMs to maintain optimal GM performance. If applications do not mark packets correctly, it is recommended to mark the traffic before the traffic reaches a GM (that is, at the access layer routers). Because Differentiated Services 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/en/US/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND/Enterprise_QoS_SRND.pdf

For a detailed description of QoS handling with GETVPN on the same device, see 4.3.4.1 “QoS.”

Figure 38. 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.

4.3 Branch Design

This section provides an overview of GETVPN 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 GETVPN 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 GETVPNGETVPN 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:

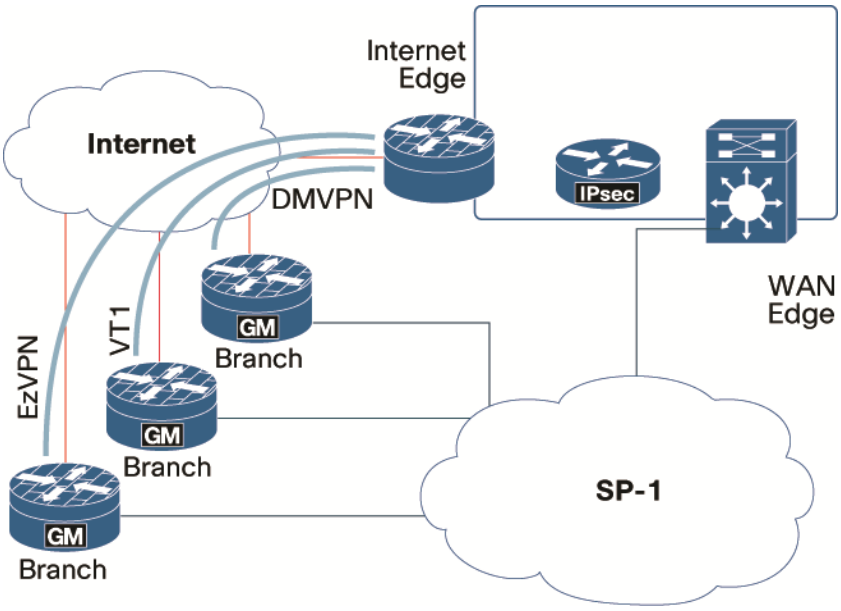
- Tier 1 branches receive a default route from the MPLS PE using Border Gateway Protocol (BGP).
- The branch receives a dynamic address and a second default route (usually with a metric of 254) from the Internet provider
- Corporate security policy does not permit split tunneling over the Internet. That 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/docs/solutions/Enterprise/Security/SAFE_RG/SAFE_rg.html

Figure 39. 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 172.19.2.1 is for DMVPN HUB
!
pre-shared-key address 172.19.2.1 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 14
  hash sha256
  lifetime 1200
!
! This policy is for DMVPN
!
crypto isakmp policy 20
```

```
  encr aes
  authentication pre-share
  group 14
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 getvpn
  identity number 61440
  server address ipv4 172.16.4.2
  server address ipv4 172.18.5.2
!
! crypto map getvpn_cm 10 gdoi
  set group getvpn
!
! 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 172.19.2.1
  ip nhrp map 172.20.1.254 172.19.2.1
  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 172.19.2.1
  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 192.168.1.1 255.255.0.0
!
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 172.19.1.2 255.255.0.0
  ip tcp adjust-mss 1360
  crypto map getvpn_cm
!
```

```
! Running EIGRP over DMVPN Tunnel
!
router eigrp 10
  network 192.168.0.0 0.0.255.255
  network 172.20.1.0 0.0.0.255
  no auto-summary

!
router bgp 106
  no synchronization
  bgp log-neighbor-changes
  network 192.168.0.0 mask 255.255.0.0
  neighbor 172.19.1.1 remote-as 500
  neighbor 172.19.1.1 update-source Serial0/0/0.102
  no auto-summary
!
! IP route for DMVPN hub pointing to DHCP
!
ip route 172.19.2.1 255.255.255.255 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 172.19.1.1 to network 0.0.0.0
```

```
172.20.0.0/24 is subnetted, 1 subnets
C   172.20.1.0 is directly connected, Tunnel1
172.21.0.0/16 is subnetted, 1 subnets
C   172.21.0.0 is directly connected, GigabitEthernet0/0
172.19.0.0/16 is subnetted, 1 subnets
C   172.19.0.0 is directly connected, Serial0/0/0.102
172.19.0.0.0/32 is subnetted, 1 subnets
S   172.19.2.1 [1/0] via 172.21.1.254
192.168.0.0/16 is subnetted, 1 subnets
C   192.168.0.0 is directly connected, GigabitEthernet0/1.203
B*  0.0.0.0/0 [20/0] via 172.19.1.1, 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
172.21.0.0/16 is subnetted, 1 subnets
C   172.21.0.0 is directly connected, GigabitEthernet0/0
172.19.0.0/16 is subnetted, 1 subnets
S   172.19.2.1 [1/0] via 172.21.1.254
192.168.0.0/16 is subnetted, 1 subnets
C   192.168.0.0 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 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 14
  lifetime 1200
!
! Policy for VTI
!
crypto isakmp policy 20
  encr aes
  authentication pre-share
  group 14
!
! Preshared key for VTI headend
!
crypto isakmp key ipsec123 address 172.19.2.1
!
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-sha256-hmac
!
crypto ipsec profile VTI
  set transform-set AES_SHA
!
crypto gdoi group dgvpn
  identity number 61440
  server address ipv4 172.16.4.2
  server address ipv4 172.18.5.2
  server address ipv4 172.15.3.2
!
crypto map dgvpn 10 gdoi
  set group dgvpn
!
! Virtual Tunnel Interface
!
interface Tunnel1
  ip address 172.20.1.1 255.255.255.0
  tunnel source GigabitEthernet0/2
  tunnel destination 172.19.2.1
  tunnel mode ipsec ipv4
  tunnel protection ipsec profile VTI
!
interface GigabitEthernet0/1
  description TO MPLS WAN
  ip address 172.15.1.2 255.255.0.0
  crypto map dgvpn
!
interface GigabitEthernet0/2
  description TO INTERNET
  ip address 172.16.1.101 255.255.255.0
!
```

```
interface FastEthernet2/0
  description TO LAN
  ip address 192.168.1.1 255.255.0.0
!
router bgp 109
  no synchronization
  bgp log-neighbor-changes
  network 192.168.0.0 mask 255.255.0.0
  neighbor 172.15.1.254 remote-as 500
  neighbor 172.15.1.254 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 172.19.2.1 255.255.255.255 172.16.1.254
```

4.3.1.2.1 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 172.15.1.254 to network 0.0.0.0

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

```
172.16.0.0/16 is subnetted, 1 subnets
C   172.16.0.0 is directly connected, GigabitEthernet0/2
172.15.0.0/16 is subnetted, 1 subnets
C   172.15.0.0 is directly connected, GigabitEthernet0/1
172.19.0.0/32 is subnetted, 1 subnets
S   172.19.2.1 [1/0] via 172.16.1.254
192.168.0.0/16 is subnetted, 1 subnets
C   192.168.0.0 is directly connected, FastEthernet2/0
B*  0.0.0.0/0 [20/0] via 172.15.1.254, 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

172.20.0.0/24 is subnetted, 1 subnets
C   172.20.1.0 is directly connected, Tunnel1
172.16.0.0/16 is subnetted, 1 subnets
C   172.16.1.0 is directly connected, GigabitEthernet0/2
172.19.0.0/32 is subnetted, 1 subnets
S   172.19.2.1 [1/0] via 172.16.1.254
192.168.0.0/16 is subnetted, 1 subnets
C   192.168.0.0 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 GETVPN 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 GETVPN 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.

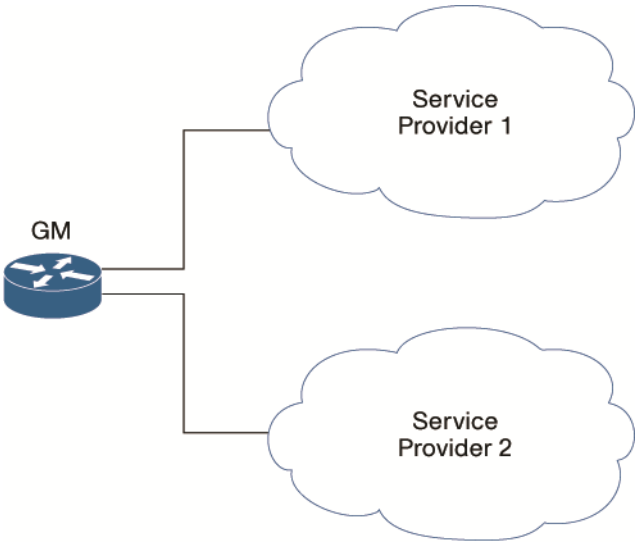
4.3.2.1 Dual WAN Termination on a Single Branch Router (GM)

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 172.16.4.2
!
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
```

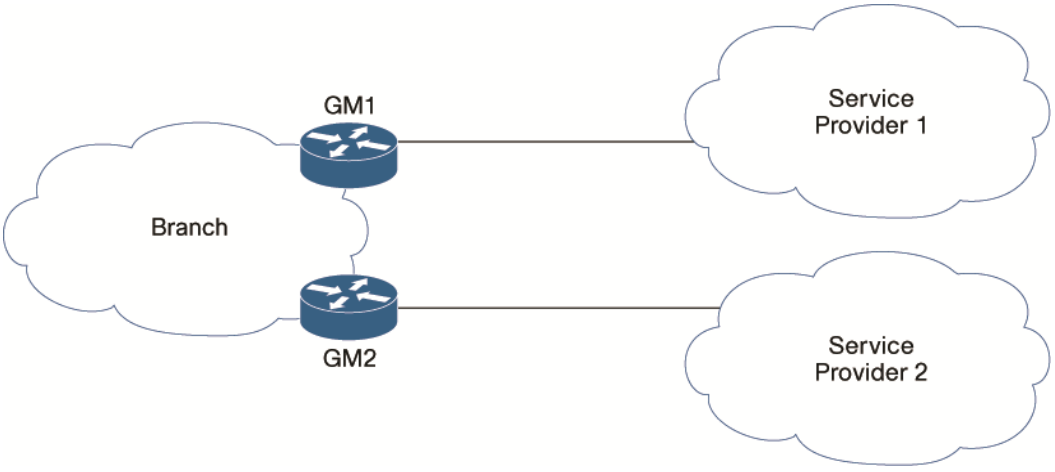
Figure 40. Dual WAN Termination on a Single Branch Router



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, GETVPN GMs share the same SAs and policies. Therefore, the branch can handle asymmetric routing.

Figure 41. Dual Tier Branch – CEs Acting as GM



A sample follows:

GM1

```
crypto gdoi group dgvpn
  identity number 61440
  server address ipv4 172.16.4.2
```

```
crypto map dgvpn 10 gdoi
  set group dgvpn

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

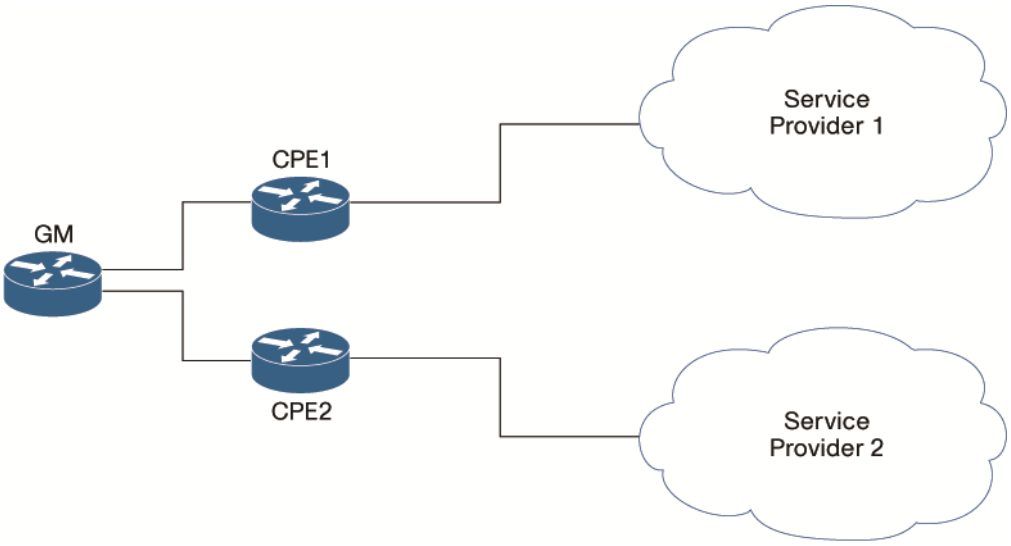
GM2

```
crypto gdoi group dgvpn
  identity number 61440
  server address ipv4 172.18.5.2

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 GETVPN on the CPEs. An enterprise-owned branch router acts as a GETVPN GM and connects to both CPEs.

Figure 42. 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 172.16.4.2
!
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 GETVPN 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 GETVPN Branch Deployment Features

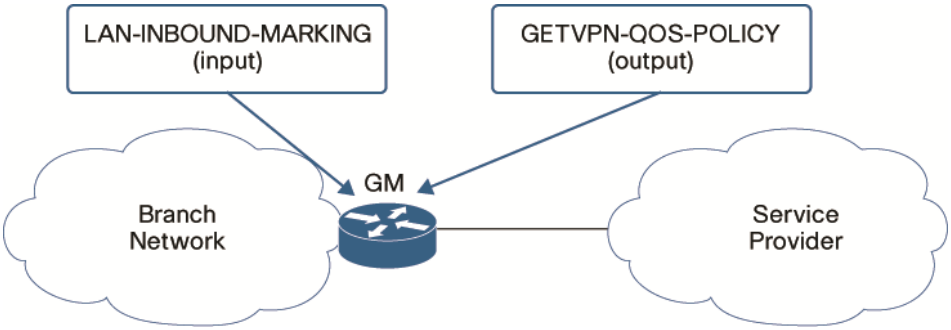
Common GETVPN branch deployment features include:

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

4.3.4.1 QoS

In GETVPN 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 GETVPN crypto map attached interface. Figure 41 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 43. 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

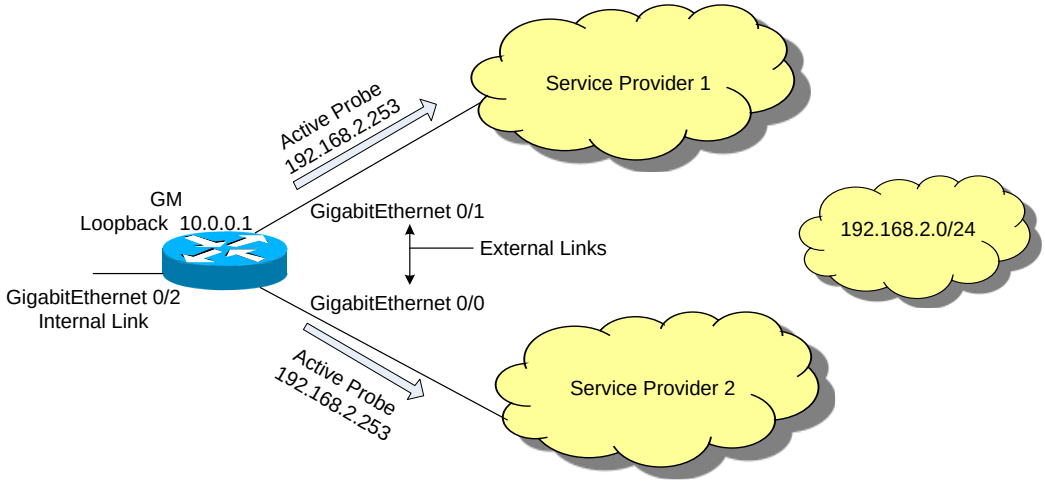
Because GETVPN dual-tier branches have multiple paths, maintaining end-to-end connectivity is vital in enterprise network operation. Cisco PfR (Performance Routing) technology can be deployed in branches with redundant links to route the traffic based on various network parameters such as delay, jitter, 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 Figure 44, 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 GETVPN functionality. PfR fast-monitoring is used for monitoring connectivity to a critical site that hosts the LAN 192.168.2.0/24. To do this, an active probe is configured for one host address (192.168.2.253) 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 44. PfR in the GETVPN 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 10.0.0.1 key-chain BR1
 interface GigabitEthernet0/2 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 10.0.0.1 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 192.168.2.253 set probe frequency 10
!
!prefix-list used by oer-map
!network 192.168.2.0/24 is Datacenter LAN network, and hence critical
ip prefix-list OER seq 5 permit 192.168.2.0/24
!
```

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

```
GM#show pfr 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
10.0.0.1  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 PfR prefixes created by the PfR master, use the command **show pfr master prefix**. Fast monitoring creates its own prefix based on the prefix-list configured for the oer-map. In our example, this would be 192.168.2.0/24.

Passive monitoring creates its prefixes based on the traffic flow. In our example, the GM was carrying traffic to a destination 192.168.6.0/24, but the aggregation-type is configured for a prefix-length of 16. Therefore, passive monitoring creates the prefix 192.168.0.0/16.

```
GM#show pfr 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


--------------------------------------------------------------------------------
192.168.0.0/16   INPOLICY      0      10.0.0.1      Gi0/0         BGP
                       U      U            0      0      0      0
                       N      N            N      N      1      1
N   N
192.168.2.0/24   INPOLICY      @0     10.0.0.1      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 PfR:

```
GM#show pfr 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
192.168.2.253    N 172.16.1.1 Gi0/1  173          171    0
Echo  192.168.2.253    N 172.15.1.1  Gi0/0  173          173    0
```

In the preceding example, PfR 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 PfR route information:

```
GM#show pfr border route bgp
BGP table version is 2366, local router ID is 172.15.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              r RIB-failure, S Stale
```

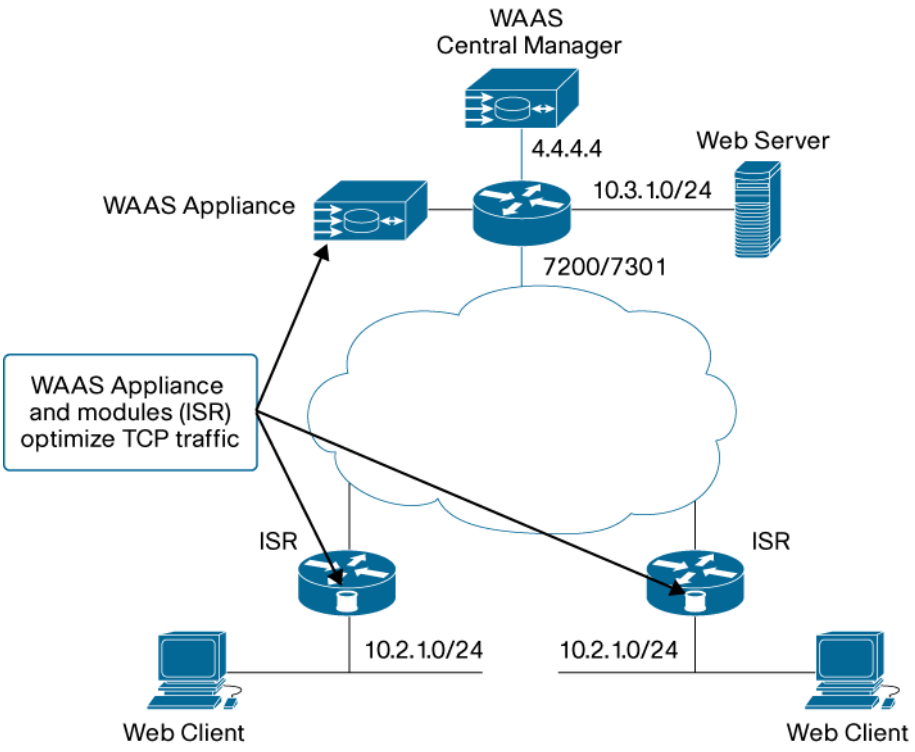
```
Origin codes: i - IGP, e - EGP, ? - incomplete
OER Flags: C - Controlled, X - Excluded, E - Exact, N - Non-exact, I -
Injected
Network    Next Hop      OER    LocPrf        Weight        Path
*> 192.168.2.0/24 172.16.1.254       CEI                  0             500 ?
!
```

4.3.4.3 WAAS

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 GETVPN environment by using a WAAS appliance or network modules (ISR routers).

To configure WAAS services on a GETVPN 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 45:

Figure 45. WAAS Deployment in a GETVPN Environment



4.3.4.3.1 Router Configuration

The following configuration is required on the router in addition to the GETVPN 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
```

```
!
```

4.3.4.4.3.2 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 GETVPN 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
```

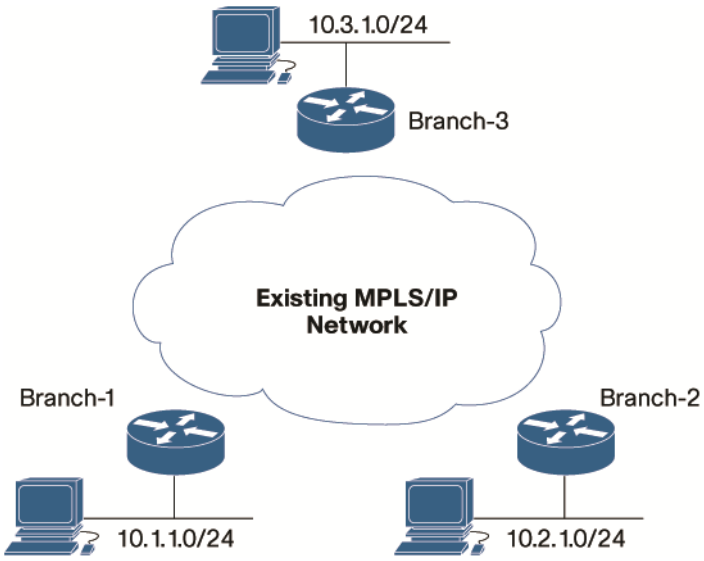
In addition to the preceding commands, **show interface** counters can compare LAN and WAN data rates. Typically, the WAN traffic rate is a fraction of LAN data rate. In addition to improving latency and response times, WAAS reduces (compresses) TCP traffic flowing through the router to improve WAN utilization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

A GETVPN network is seldom deployed in a new customer deployment. Typically a GETVPN 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 46. 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 an 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 GETVPN Deployment

In GETVPN 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 GETVPN 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 GETVPN Operating Modes

GETVPN 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 GETVPN to come up with a successfully migration strategy. Since the encryption and decryption behavior of the GETVPN devices is different in each mode, Table 7 summarizes this behavior:

Table 5. GETVPN Operating Modes

Mode	Traffic direction with respect to the GM	• Behavior
Conformant (Normal)	Receiving (Incoming from the WAN)	• 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 (Outgoing to the WAN)	• Clear traffic not matching policy is forwarded in clear • Clear traffic matching policy is encrypted
Receive Only - Configured on KS and applicable to all GMs registering to this Group	Receiving (Incoming from the WAN)	• 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 (Outgoing to the WAN)	• Clear traffic matching policy is **forwarded** • Clear traffic not matching policy is forwarded
Passive Mode - Configured on individual GM and will override the receive-only option pushed by KS	Receiving (Incoming from the WAN)	• 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 (Outgoing to the WAN)	• Clear traffic matching policy is encrypted • Clear traffic not matching policy is forwarded

4.4.4 Migration Strategy: Receive-Only SAs

Migration to a GETVPN 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 GETVPN to be deployed using the phased approach, a few devices at a time. Devices that have been configured as GETVPN 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 GETVPN 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 172.168.4.2 with seq # 1

GM#show crypto 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#show crypto gdoi

GROUP INFORMATION

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

4.4.5 Migration Strategy – Testing before Deploying

4.4.5.1 Receive-only migration strategy

GETVPN 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 GETVPN 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
```

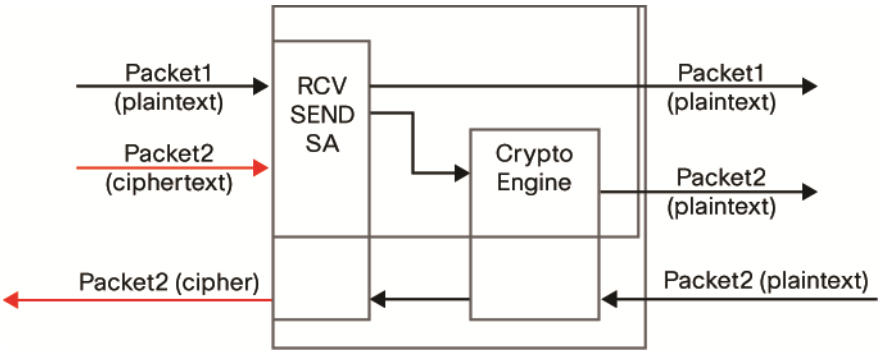
Note: Rekey from the KS overrides user-defined IPsec SA direction settings to the KS configured value. Users must to reissue the command to get the desired setting.

4.4.5.2 Passive Mode Migration Strategy

Passive Mode is useful for migration from a non-IPsec environment to an IPsec environment in a contolled fashion. The GM is in Passive Mode as a transient state after which it will switch to normal (conformant) mode.

The figure and statements below describe what happens to data traffic that hits a group member that is configured to be in Passive Mode.

Figure 47. Passive Mode of Operation



On the inbound side,

- If packets are in the clear and don't match any ACE, the packets are forwarded.
- If packets are encrypted and match the SPI, the packets are decrypted and forwarded.
- If packets are encrypted but don't match the SPI, the packets are dropped.

On the outbound side,

- If there is no match on the clear packet, the packet is forwarded.
- If there is a match on the deny, the packet is sent in the clear.
- If there is a match on the permit and there is an SA, the packet is encrypted and sent.

If there is a match but there is no SA, the packet is dropped.

5. Provisioning, Verification, and Monitoring

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

5.1 Deploying GETVPN using CSM

Cisco Security Manager (CSM) can be used to deploy and manage Group Member (GM) and Key Server (KS) configurations efficiently. The powerful “Flex Configuration” feature of the CSM can be used for GETVPN 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.

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

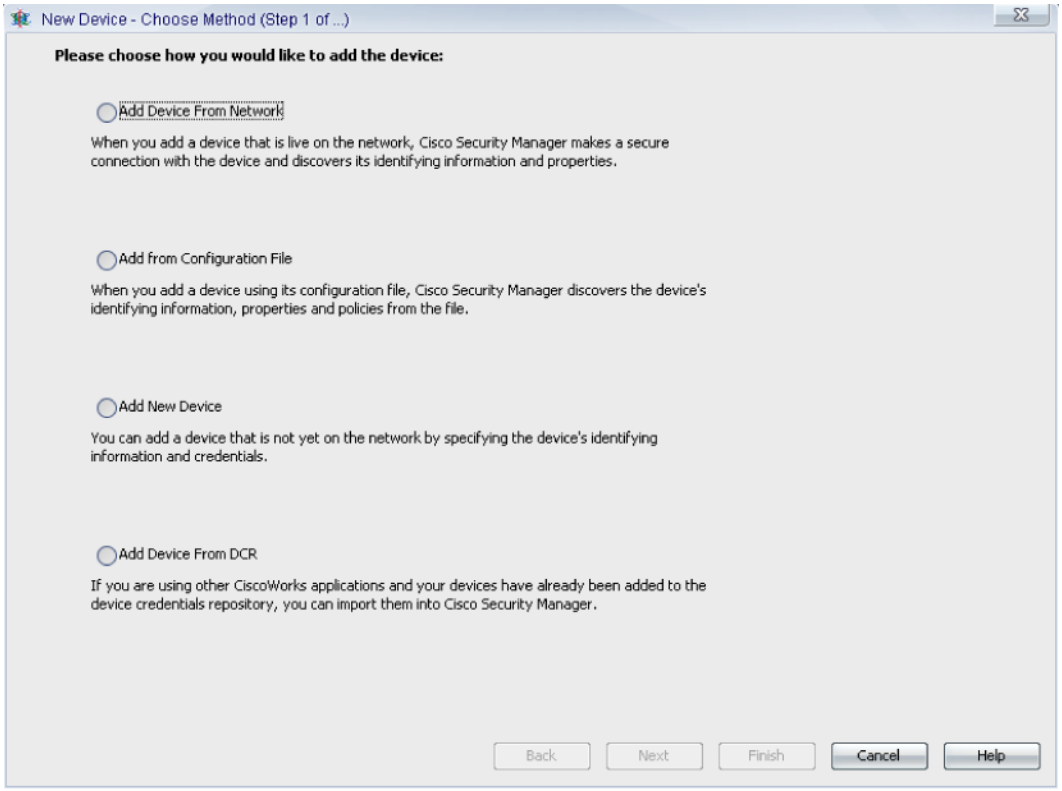
Step 1. Configure IOS Device to accept CSM connections:

- 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 48. 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 aes
 authentication pre-share
 group 14
 hash sha256
!
```

```
crypto isakmp key cisco123 address 0.0.0.0 0.0.0.0
!
!
crypto ipsec transform-set aes_sha esp-aes 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 172.16.4.2
  redundancy
    local priority 100
    peer address ipv4 172.18.5.2
!
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 GETVPN 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.

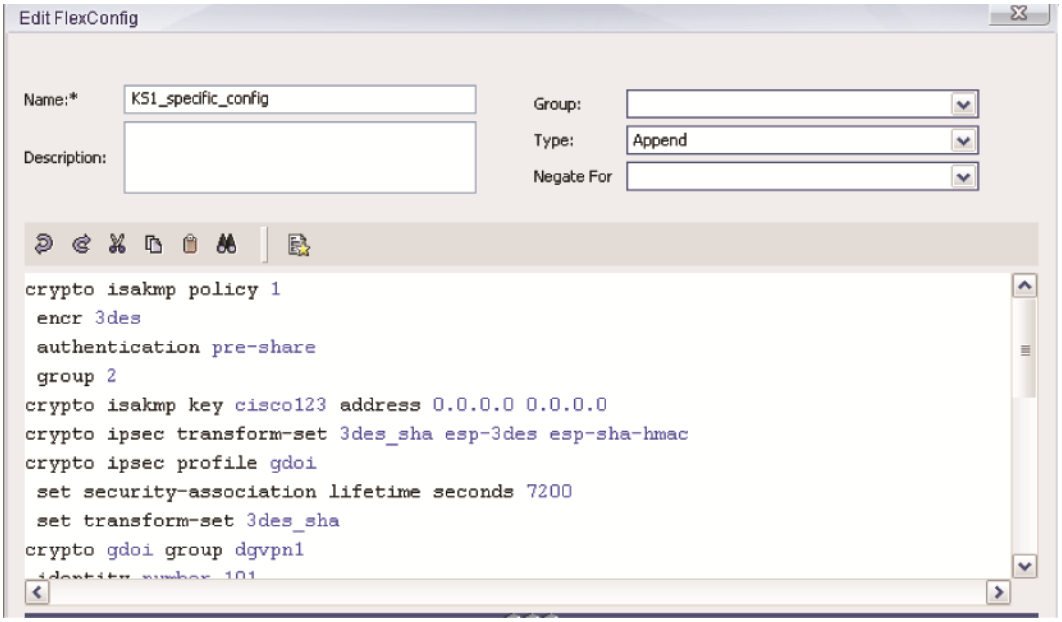
Because any changes made to the shared Flex Configuration must be deployed to all devices sharing the configuration, the chances of inconsistency between KS policies are reduced using this deployment method. This deployment method also helps overcome the inherent issue of configuration sync between the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

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

5.1.1.1 Adding a KS Specific FlexConfig

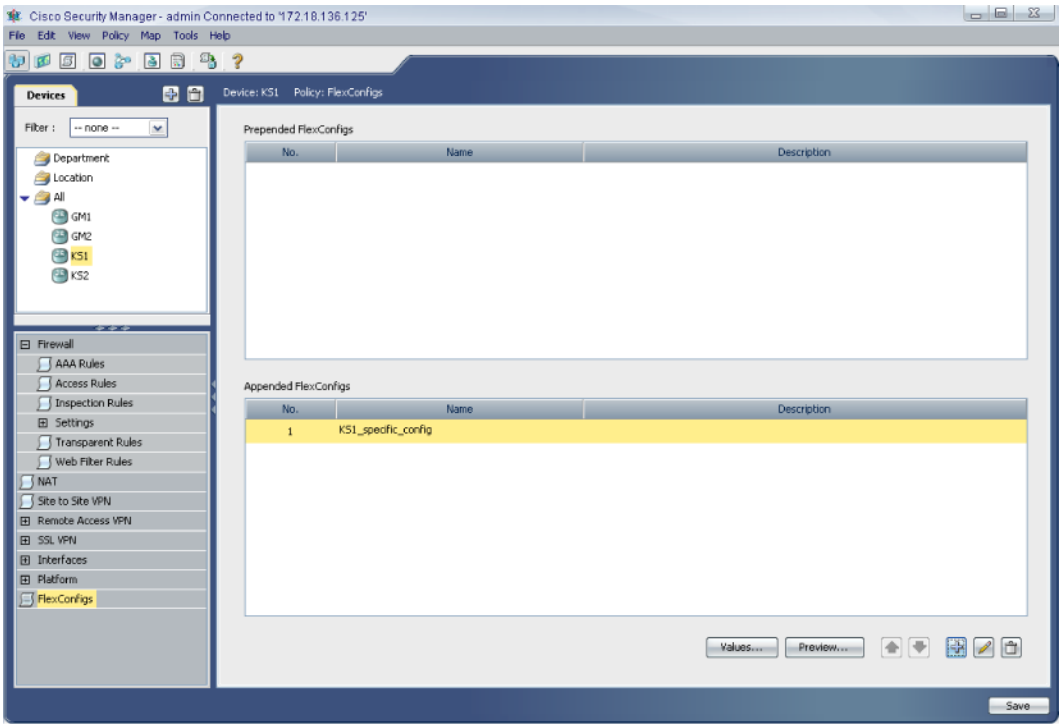
1. Click the KS device icon.
2. Click FlexConfigs.
3. Click the + button to add a new FlexConfig.
4. In the **FlexConfig** selector menu box, click + to add a new FlexConfig. Give this configuration a descriptive name, such as **KS1_specific_config**.
5. Add the **complete** KS configuration to the text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

Figure 49. 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 50. 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 172.18.5.2
!
crypto gdoi group dgvpn1
 identity number 101
 server local
  address ipv4 172.16.4.2
  redundancy
   local priority 100
   peer address ipv4 172.18.5.2
!
```

For CSM deployment:

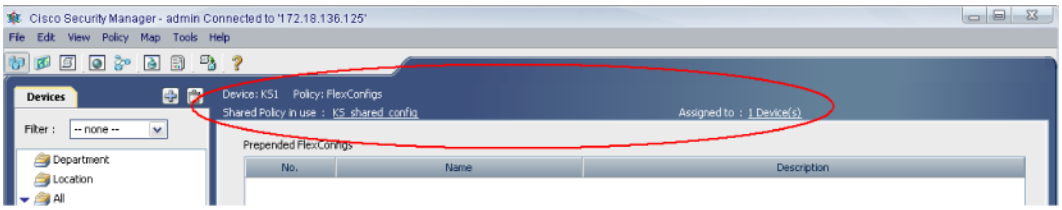
1. Click the KS device icon
2. Click FlexConfigs
3. Click the + button to add a new FlexConfig

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 51. 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

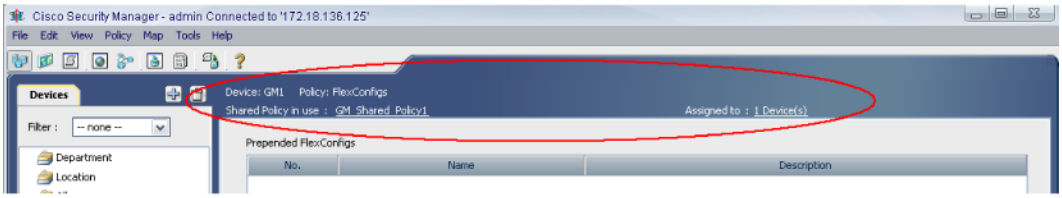
Deploying the group member (GM) configuration using CSM is relatively simple. GM configuration can be deployed by creating a FlexConfig and then sharing the configuration amongst all the GMs.

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 172.16.4.2
crypto isakmp key cisco456 address 172.18.5.2
!
crypto gdoi group dgvpn1
 identity number 101
 server address ipv4 172.16.4.2
 server address ipv4 172.18.5.2
!
!
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 52. 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 172.16.4.2 (preferred)
server address ipv4 172.18.5.2
```

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

```
server address ipv4 172.18.5.2 (preferred)
server address ipv4 172.16.4.2
```

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

5.2 GETVPN Syslog Capabilities

Syslog messages provide a valuable resource for monitoring and troubleshooting GETVPN. 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 8 lists and describes syslog messages associated with GETVPN.

Table 6. GETVPN Syslog Messages

Message	Explanation
COOP_CONFIG_MISMATCH	The configuration between the primary KS and secondary KS are mismatched.
COOP_KS_ADD	A KS has been added to the list of cooperative KSs in a group.

COOP_KS_ELECTION	The local KS has entered the election process in a group.
COOP_KS_REACH	The reachability between the configured cooperative KSs is restored.
COOP_KS_REMOVE	A KS has been removed from the list of cooperative KSs in a group.
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. Could be considered a hostile event.
COOP_KS_UNREACH	The reachability between the configured cooperative KSs is lost. Could be considered a hostile event.
COOP_KS_VER_MISMATCH	KSs are running different versions of the Cisco IOS code.
COOP_PACKET_DROPPED	A hard limit set on the driver buffer size prevents the sending of packets this size or larger.
GDOI-3-GDOI_REKEY_SEQ_FAILURE	The rekey message is rejected because the sequence number antireplay check failed.
GDOI-3-GM_NO_CRYPTO_ENGINE	No crypto engine is found due to a lack of resources or an unsupported feature requested.
GDOI-3-PSEUDO_TIME_LARGE	The rekey has a larger pseudotime that exceeds the calculated allowable pseudotime difference.
GDOI-3-PSEUDO_TIME_TOO_OLD	The rekey has a smaller pseudotime that exceeds the calculated allowable pseudotime difference.
GDOI-4-GDOI_ANN_TIMESTAMP_ LARGE	The secondary KS receives from the primary KS an ANN that has a larger pseudotime that exceeds the calculated allowable pseudotime difference.
GDOI-4-GDOI_ANN_TIMESTAMP_ TOO_OLD	The secondary KS receives from the primary KS an ANN that has a smaller pseudotime that exceeds the calculated allowable pseudotime difference.
GDOI-5-COOP_KS_BLOCK_NEW_GM_REGISTER	The secondary KS temporarily blocks a GM from registering in a group because it has not received a valid pseudotime from the primary KS.
GDOI-5-COOP_KS_VALID_ANN_ TIMER_EXPIRED	The secondary KS keeps receiving ANNs with invalid pseudotimes after three retransmits. The secondary KS temporarily blocks new group-member registration until a valid ANN is received.
GDOI_ACL_NUM	The ACL has too many entries. GDOI will honor only the first 100 ACL entries specified.
GDOI_REKEY_FAILURE	During GDOI rekey the payload parsing failed on this GM from the KS.
GM_ACL_MERGE	The ACL differences between a GM and KS are resolved and a merge took place.
GM_ACL_PERMIT	The GM can support only an ACL for "deny." Any traffic matching the "permit" entry will be dropped.
GM_CLEAR_REGISTER	The clear crypto gdoi command has been executed by the local GM.
GM_CM_ATTACH	A crypto map has been attached for the local GM.
GM_CM_DETACH	A crypto map has been detached for the local GM.
GM_CONV_SA_DUPLEX	IPsec SAs have been converted to bidirectional mode in a group on a GM.
GM_CONV_SA_DUPLEX_LOCAL	IPsec SAs have been converted to bidirectional mode in a group on a GM by a CLI command.
GM_ENABLE_GDOI_CM	A GM has enabled ACL on a GDOI crypto map in a group with a KS.
GM_HASH_FAIL	During GDOI registration protocol, a message sent by the KS has bad or no hash.
GM_INCOMPLETE_CFG	Registration cannot be completed because the GDOI group configuration may be missing the group ID, server ID, or both.
GM_RE_REGISTER	The IPsec SA created for one group may have been expired or cleared. Need to re-register to the KS.
GM_RECV_DELETE	A message sent by the KS to delete the GM has been received.

GM_RECV_REKEY	Rekey received.
GM_REGS_COMPL	Registration complete.
GM_REJECTING_SA_PAYLOAD	During GDOI registration protocol, a proposal sent by the KS was refused by the local GM.
GM_REKEY_IPV4_POLICY_CHECK_FAIL	A GM that is configured to join an IPv4 group has mistakenly received an IPv6 policy during a rekey. This might be because of a configuration error on the KS in which the same group is configured with an IPv6 policy.
GM_REKEY_IPV6_POLICY_CHECK_FAIL	A GM that is configured to join an IPv6 group has mistakenly received an IPv4 policy during a rekey. This might be because of a configuration error on the KS in which the same group is configured with an IPv4 policy.
GM_REKEY_NOT_RECD	A GM has not received a rekey message from a KS in a group. Currently unimplemented.
GM_REKEY_TRANS_2_MULTI	A GM has transitioned from using a unicast rekey mechanism to using a multicast mechanism.
GM_REKEY_TRANS_2_UNI	A GM has transitioned from using a multicast rekey mechanism to using a unicast mechanism.
GM_SA_INGRESS	A received-only ACL has been received by a GM from a KS in a group.
GM_UNREGISTER	A GM has left the group.
KS_BAD_ID	A configuration mismatch exists between a local KS and a GM during GDOI registration protocol.
KS_BLACKHOLE_ACK	A KS has reached a condition of black-holing messages from a GM. Could be considered a hostile event.
KS_CLEAR_REGISTER	The clear crypto gdoi command has been executed by the local KS.
KS_CONV_SAS_DUPLEX	IPsec SAs have been converted to bidirectional mode in a group.
KS_CONV_SAS_INGRESS	IPsec SAs have been converted to receive-only mode in a group.
KS_FIRST_GM, GDOI, LOG_INFO	A local KS has received the first GM joining the group.
KS_GM_REJECTS_SA_PAYLOAD	During GDOI registration protocol, a proposal sent by the KS was refused by the GM.
KS_GM_REVOKED	During rekey protocol, an unauthorized member tried to join a group. Could be considered a hostile event.
KS_GROUP_ADD	A configuration command has been executed to add a KS in a group.
KS_GROUP_DELETE	A configuration command has been executed to remove a KS from a group.
KS_HASH_FAIL	During GDOI registration protocol, a message sent by the GM has a bad or no hash.
KS_LAST_GM	The last GM has left the group on the local KS.
KS_NACK_GM_EJECT	The KS has reached a condition of not receiving an ACK message from a GM and has been ejected.
KS_NO_RSA_KEYS	RSA keys were not created or they are missing.
KS_REGS_COMPL	The KS has successfully completed a registration in a group.
KS_REKEY_TRANS_2_MULTI	The group has transitioned from using a unicast rekey mechanism to a multicast mechanism.
KS_REKEY_TRANS_2_UNI	The 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.

KS_UNSOL_ACK	The KS has received an unsolicited ACK message from a past GM or is under a DOS attack. 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 antireplay 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.
UNEXPECTED_SIGKEY	An unexpected signature key was found that frees the signature key.
UNREGISTERED_INTERFACE	Receiving registration from unregistered interface. Stop processing it.
UNSUPPORTED_TEK_PROTO	Unexpected TEK protocol.

5.3 GDOI Event Trace

The GDOI Event Tracing feature provides a trace facility for troubleshooting GETVPN. This feature enables monitoring of GETVPN events, errors, and exceptions. During runtime, the event trace mechanism logs trace information in a buffer space. A display mechanism extracts and decodes the debug data.

The GDOI Event Tracing feature can be used to analyze the cause of a device failure. When the GDOI Event Tracing feature is configured, the router logs messages from specific GETVPN subsystem components into the device memory, which can be viewed as trace messages that are stored in the memory or saved to a file.

How to Configure GDOI Event Tracing

GDOI Event Tracing feature can be configured either in privileged EXEC mode or global configuration mode based on the desired parameters. The following command is used to configure GDOI event tracing.

```
  monitor event-trace gdoi  {coop | infra | registration | rekey}
```

Following is a snapshot of the gdoi event-trace options that are configurable:

```
GM1#monitor event-trace gdoi ?
  coop          GDOI COOP Event Traces
  dump          Dump all the event traces
  exit          GDOI Exit Traces
  infra         GDOI INFRA Event Traces
  registration  GDOI Registation event Traces
  rekey         GDOI rekey event Traces
```

How to display information collected by GDOI event trace

Following command shows the options available to display the information collected by the gdoi event monitor:

```
GM1#show monitor event-trace gdoi?
  all           Show all the traces in current buffer
  back          Show trace from this far back in the past
  clock         Show trace from a specific clock time/date
  coop          GDOI COOP Event Traces
  from-boot     Show trace from this many seconds after booting
  infra         GDOI INFRA Event Traces
```

```
  latest        Show latest trace events since last display
  merged        Show entries in all event traces sorted by time
  registration  GDOI Registration event Traces
  rekey         GDOI Rekey event Traces
```

Example output of an event-trace enabled on a GM:

```
GM1#show monitor event-trace gdoi merged all
*May      25      20:20:57.706:      Registration_events:      GDOI_REG_EVENT:
REGISTRATION_STARTED:
                                          GM  10.1.20.2  to  KS  10.1.11.2  for
group G1
*May  25  20:21:08.970:  Registration_events:  GDOI_REG_EVENT:  REGISTRATION_DONE:
GM
                                          10.1.13.2 to KS 10.1.11.2 for group
G1
*May   26   00:45:52.878:   Rekey_events:   GDOI_REKEY_EVENT:   REKEY_RCVD:   From
10.1.11.2
                                    to 10.1.13.2 with seq no 131 for the group
G1
*May 26 00:45:52.878: Rekey_events: GDOI_REKEY_EVENT: ACK_SENT: From 10.1.11.2
                                   to 10.1.13.2 with seq no 131 for the group
G1
```

5.4 GETVPN Verification

GETVPN 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.

5.4.1 Verifying KS Operation

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

5.4.1.1 show crypto gdoi

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

```
Primary_KS#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          : 172.16.4.2
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
```

5.4.1.2 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 192.168.0.0 0.0.255.255 192.168.0.0 0.0.255.255
access-list      199    deny ip any any
```

5.4.1.3 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.

5.4.1.3.1 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
```

5.4.1.3.2 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
```

5.4.1.4 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 GETVPN operation. A rekey SA (shown as GDOI_REKEY) however always stays in the IKE database.

```
Primary_KS#sh crypto isakmp sa
IPv4 Crypto      ISAKMP SA
dst        src          state        conn-id      slot   status
172.16.4.2 192.168.1.5 GDOI_IDLE    1002         0      ACTIVE
172.16.4.2 10.1.1.9    GDOI_IDLE    1003         0      ACTIVE
172.16.4.2 10.1.1.13   GDOI_IDLE    1004         0      ACTIVE
10.1.1.13  172.16.4.2  GDOI_REKEY   0            0      ACTIVE
```

Note: When a primary KS is unconfigured from the GM, the rekey ISAKMP SA associated with the KS will be deleted from the GM. The GM will reregister with a KS in the order of the configuration.

5.4.1.5 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.

5.4.1.5.1 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  172.16.4.2 192.168.1.5   ACTIVE aes sha psk 2 23:55:17
      Engine-id:Conn-id = SW:2

1003  172.16.4.2 10.1.1.9      ACTIVE aes sha psk 2 23:55:31
      Engine-id:Conn-id = SW:3

1004  172.16.4.2 10.1.1.13     ACTIVE aes sha psk 2 23:55:43
      Engine-id:Conn-id = SW:4
```

5.4.1.5.2 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  172.16.4.2 192.168.1.5   ACTIVE aes sha rsig 2 23:58:12
      Engine-id:Conn-id = SW:7

1008  172.16.4.2 10.1.1.9      ACTIVE aes sha rsig 2 23:58:22
      Engine-id:Conn-id = SW:8
```

```
1009  172.16.4.2 10.1.1.13     ACTIVE aes sha rsig 2 23:58:32
```

5.4.1.6 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  : 10.1.1.9
Group ID         : 1234
Group Name       : getvpn
Key Server ID    : 172.16.4.2
Rekeys sent      : 1

Sent seq num:    1      0      0      0
Rcvd seq num:    1      0      0      0
Group Member ID  : 10.1.1.13
Group ID         : 1234
Group Name       : getvpn
Key Server ID    : 172.16.4.2
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 192.168.1.1 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 | begin 192.168.1.1
Group Member ID  : 192.168.1.1
Group ID         : 61440
Group Name       : dgvpn1
Key Server ID    : 172.16.4.2
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
```

5.4.1.7 show crypto gdoi ks coop

To display information about COOP KSs in a GETVPN environment, use 'show crypto gdoi ks coop' command. This command provides detailed information about the priorities and roles of various COOP devices.

```
Primary_KS#sh crypto gdoi ks coop
Crypto Gdoi Group Name:getvpn
Group handle: 2147483652, Local Key Server handle: 2147483652

Local Address: 172.16.4.2
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: 172.18.5.2
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
```

5.4.2 Verifying GM Operation

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

5.4.2.1 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#show crypto isakmp sa

IPv4 Crypto ISAKMP SA

dst         src         state        conn-id slot status

10.1.1.9    172.16.4.2  GDOI_REKEY   2060   0 ACTIVE

172.16.4.2  10.1.1.9    GDOI_IDLE    4038   0 ACTIVE
```

5.4.2.2 show crypto gdoi

To display information about GETVPN configuration on the GM, use the show crypto **gdoi** command on the GM. This command provides a summary of all the useful GETVPN parameters.

```
GroupMember-1#show crypto gdoi

GROUP INFORMATION


Group Name               : getvpn

Group Identity    : 1234

Rekeys received   : 1

IPSec SA Direction       : Both

Active Group Server      : 172.16.4.2

Group Server list        : 172.16.4.2
```

```
  172.18.5.2

GM Reregisters in       : 618 secs
Rekey Received(hh:mm:ss): 00:03:37

Rekeys received
   Cumulative           : 1
   After registration   : 1
Reke ’ Acks sent  : 1

ACL Downloaded From KS 172.16.4.2:
   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
```

5.4.2.3 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.

5.4.2.3.1 Multicast Rekey

```
GroupMember-1#show 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  172.16.4.2  2063    3E281825     0F770714
Current: 239.77.77.77  172.16.4.2  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     10.1.1.9
224.0.1.40       FastEthernet0/1     3w1d   00:02:10     10.1.1.9
```

5.4.2.3.2 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 :       10.1.1.9      172.16.4.2    2065   CEC03E80      F3D7CB96
Current:    10.1.1.9      172.16.4.2    2065   CEC03E80      F3D7CB96
Previous: ---     ---     ---     ---     ---
```

5.4.3 KS and GM Syslog Messages

This section describes various syslog messages that can be used to verify GETVPN operation.

5.4.3.1 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 172.16.4.2 for group dgvpn
using address 192.168.1.1
```

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 172.16.4.2 complete for group
dgvpn using address 192.168.1.1
```

5.4.3.2 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 GETVPN 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 172.16.4.2 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 GETVPN 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 172.16.4.2 with seq # 1
```

```
%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group getvpn from
address 172.16.4.2 with seq # 2
```

```
%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group getvpn from
address 172.16.4.2 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 172.16.4.2 with seq # 9

%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group getvpn from address 172.16.4.2 with seq # 12
```

5.4.3.3 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 192.168.1.1 received a rekey with sequence # 5 from the primary KS 172.16.4.2

```
%GDOI-5-GM_RECV_REKEY: Received Rekey for group dgvpn from 172.16.4.2 to 192.168.1.1 with seq # 5
```

5.4.3.4 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 172.16.4.2 for group dgvpn using address 192.168.1.1
```

5.4.3.5 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 registration as interface GigabitEthernet0/1 is down
```

5.4.3.6 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
```

5.4.3.7 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 10.1.1.9 deleted from group getvpn.
```

5.4.4 COOP Syslog Messages

This section describes various syslog messages that can be used to verify COOP operation in a GETVPN environment.

5.4.4.1 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)
```

5.4.4.2 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 172.16.4.2 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 172.18.5.2 in group getvpn transitioned to
Primary (Previous Primary = 172.16.4.2)
```

5.4.4.3 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 172.18.5.2 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 172.18.5.2
in group getvpn.
```

5.5 SNMP Monitoring using GDOI MIB

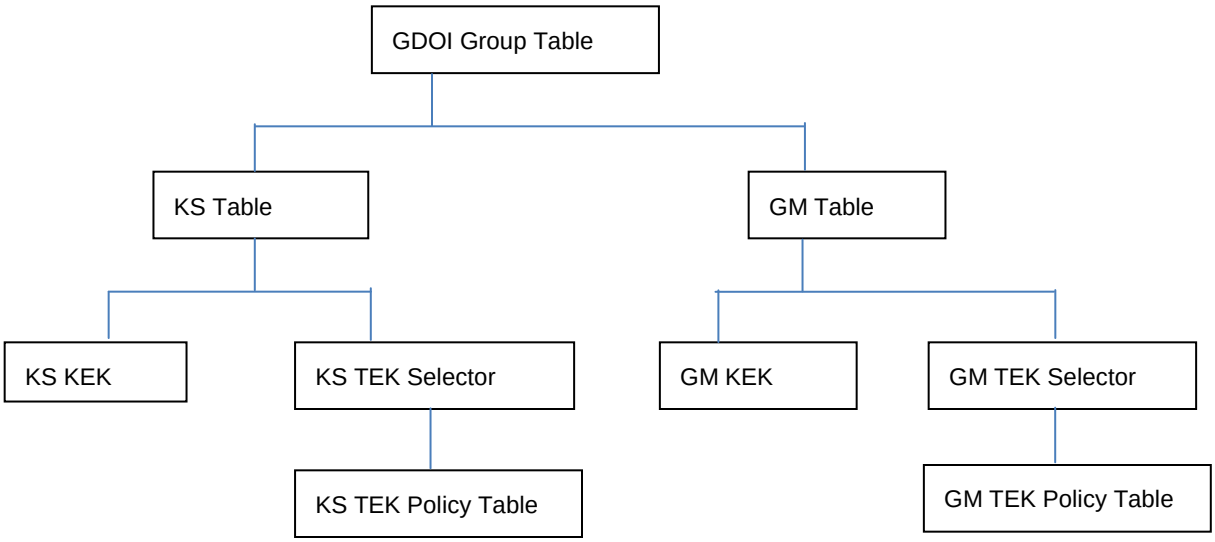
The CISCO-GDOI-MIB file can be imported into an SNMP management station and parsed to retrieve the table objects and hierarchy information through the GDOI MIB Support for GETVPN feature. The XE 3.16 GETVPN GDOI/COOP MIBS feature enhances the CISCO-GDOI-MIB via additional SNMP MIB objects for key server parameters and traps for key server events. Thes are applicable to both GDOI and G-IKEv2

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.

The GDOI MIB support is available in IOS from 15.2(1)T and IOS XE from 15.3(1)S/XE3.8. The MIB has objects that correspond to the GDOI RFC 3547. The MIB objects and notifications include information about GDOI groups, GM/KS peers, as well as the policies that are created or downloaded. The object identifier (OID) for GDOI MIB is 1.3.6.1.4.1.9.9.759. Current implementation supports only “get” operations and traps/notifications.

5.5.1 GDOI MIB Table Hierarchy

The MIB objects are categorized first by GDOI Group and then by peer type (KS or GM). Each peer has KEK and TEK tables below it. Following is the hierarchy:



GDOI MIB Table Objects

Following is a list of the MIB table objects (listed per group).

Group table objects:

- Group ID type--Specifies whether the group ID is an IP address, group number, hostname, and so on.
- Group ID length--Number of octets in the group ID value.
- Group ID value--Group number, IP address, or hostname.
- Group name--String value.

KS table objects:

- KS ID type
- KS ID length
- KS ID value
- Active KEK--SPI of the KEK that is currently used by the KS to encrypt the rekey message.
- Last rekey sequence number--Last rekey number that was sent by the KS to the group.

GM table objects:

- GM ID type
- GM ID length
- GM ID value
- Registered KS ID type--ID type of the KS to which the GM is registered.
- Registered KS ID length
- Registered KS ID value
- Active KEK--SPI of the KEK currently used by the GM to decrypt rekey messages.
- Last rekey seq number--Last rekey number received by the GM.

KS KEK table objects:

- KEK index
- KEK SPI
- KEK source ID information--Source ID type, ID length, and ID value.
- KEK source ID port--Port associated with the source ID.
- KEK destination ID information--Destination ID type, ID length, and ID value.
- KEK destination ID port--Port associated with the destination ID.
- IP protocol ID--UDP or TCP.
- Key management algorithm (unused).
- Encryption algorithm and key length (bits)
- SIG payload hash algorithm, SIG payload signature algorithm, and SIG payload key length (bits).
- Hash algorithm (will be reused from the IPsec MIB)
- Diffie-Hellman group
- KEK original lifetime (seconds)--Maximum time for which a KEK is valid.
- KEK remaining lifetime (seconds)

KS TEK selector table objects (corresponds to the ACLs that are configured as part of the IPsec SA in the GDOI group configuration on the KS):

- Tek selector index--An integer index.
- TEK source ID information--Source ID type, ID length, and ID value.
- TEK source ID port--Port associated with the source ID.
- TEK destination ID information--Destination ID type, ID length, and ID value.
- TEK destination ID port--Port associated with the destination ID.
- TEK Security protocol--GDOI_PROTO_IPSEC_ESP protocol ID value in the SA TEK payload (see RFC 3547).

KS TEK policy table objects:

- TEK policy index--An integer index.
- TEK SPI--Four octets

- Encapsulation mode--Tunnel or transport.
- Encryption algorithm and key length (bits)
- Integrity and authentication algorithm and key length (bits)
- TBAR window size (seconds)
- TEK original lifetime (seconds)--Maximum time for which a TEK is valid.
- TEK remaining lifetime (seconds)
- TEK Status--Inbound, outbound, or not in use.

GM KEK table objects:

- KEK index--An integer index.
- KEK SPI
- KEK source ID information--Source ID type, ID length, and ID value.
- KEK source ID port--Port associated with the source ID.
- KEK destination ID information--Destination ID type, ID length, and ID value.
- KEK destination ID port--Port associated with the destination ID.
- IP protocol ID--UDP or TCP.
- Key management algorithm (unused)
- Encryption algorithm and key length (bits)
- SIG payload hash algorithm, SIG payload signature algorithm, and SIG payload key length (bits)
- Hash algorithm
- Diffie-Hellman group
- KEK original lifetime (seconds)--Maximum time for which a KEK is valid.
- KEK remaining lifetime (seconds)

GM TEK selector table objects (corresponds to the ACLs that are downloaded to the GM as part of the TEK policy from the KS):

- TEK selector index--An integer index.
- TEK source ID information--Source ID type, ID length, and ID value.
- TEK source ID port--Port associated with the source ID.
- TEK destination ID information--Destination ID type, ID length, and ID value.
- TEK destination ID port--Port associated with the destination ID.
- TEK Security protocol--GDOI_PROTO_IPSEC_ESP protocol ID value in the SA TEK payload (see RFC 3547).

GM TEK policy table objects:

- TEK policy index--An integer index.
- TEK SPI --Four octets.
- Encapsulation mode--Tunnel or transport.
- Encryption algorithm and key length (bits)
- Integrity and authentication algorithm and key length (bits)
- TBAR window size (seconds)
- TEK original lifetime (seconds)--Maximum time for which a TEK is valid.
- TEK remaining lifetime (seconds)
- TEK Status--Inbound, outbound, or not in use.

GDOI MIB Traps/Notifications:

The GDOI MIB contains 2 types of Traps/Notifications: those generated by the KS and those generated by each GM. The following table has the detailed information:

Table 7. SNMP Notifications Supported by the GDOI MIB

Notification	Description
KS New Registration	A KS first received a registration request from a GM.
KS Registration Complete	A GM completed registration to the KS.
KS Rekey Pushed	A rekey message was sent by the KS.
KS No RSA Keys	An error notification was received from the KS because of missing RSA keys.
GM Register	A GM first sent a registration request to a KS.
GM Registration Complete	A GM completed registration to a KS.
GM Re-Register	A GM began the re-registration process with a KS.
GM Rekey Received	A rekey message was received by a GM.
GM Incomplete Config	A GM sent an error notification because of a missing configuration.
GM Rekey Failure	A GM sent an error notification because it cannot process and install a rekey.
KS Role Change	A KS switches between primary and secondary role
KS GM Deleted	Generated when a GM is deleted from the KS.
KS Peer Reachable	Generated by a KS when unreachable COOP peer becomes reachable.
KS Peer Unreachable	Generated by a KS when reachable COOP peer becomes unreachable.

GDOI MIB Limitations

The GDOI MIB contains only objects that are listed in RFC 3547 and does not contain objects for functionality specific to the Cisco implementation of GDOI. This functionality includes:

- Cooperative key servers
- GM ACLs
- Receive-only SAs
- Fail-close/fail-open
- Crypto map objects
- Other Cisco GETVPN specific features

5.5.1 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 7 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 http://www.cisco.com

Table 8. 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)

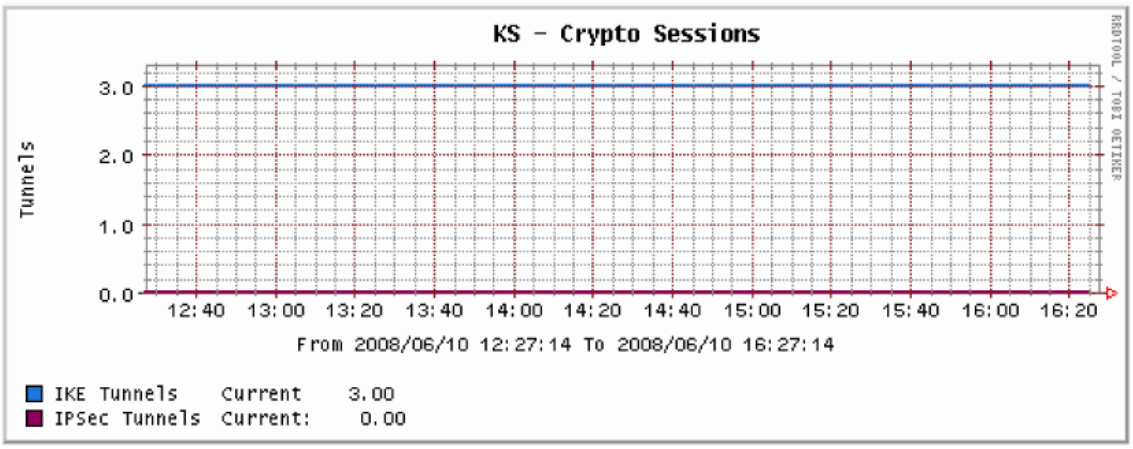
The following sections describe how to use SNMP plots to analyze GET VPN network behavior.

5.5.1.1 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.

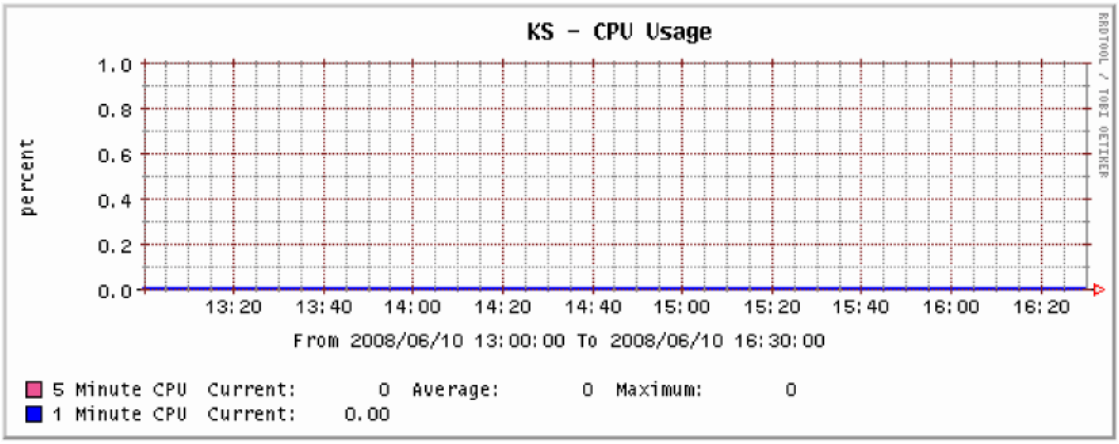
Figure 44 shows a sample plot obtained by SNMP polling the KS. In case of four COOP KSs, each KS maintains 3 IKE tunnels.

Figure 46. 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 47. SNMP Polling: Steady state – CPU

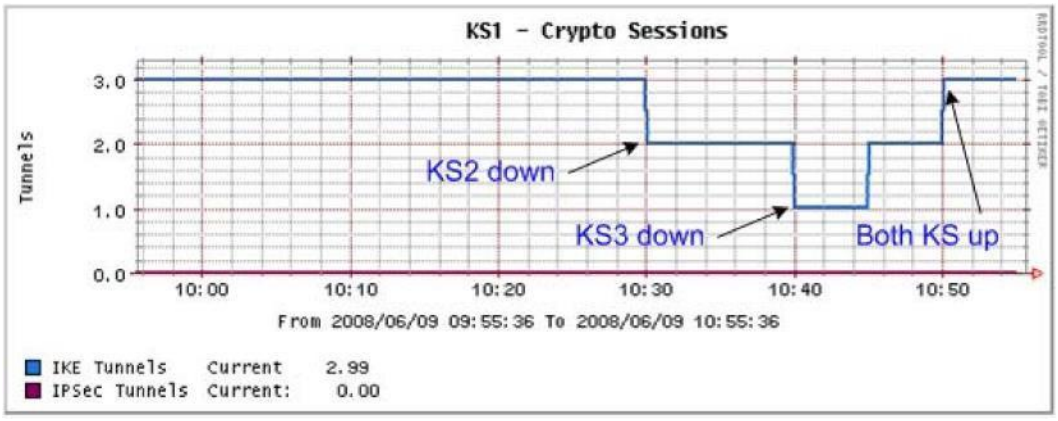


Note: A CPU spike is seen at registration time and during rekey, as shown in Table 2.

5.5.1.2 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 46 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.

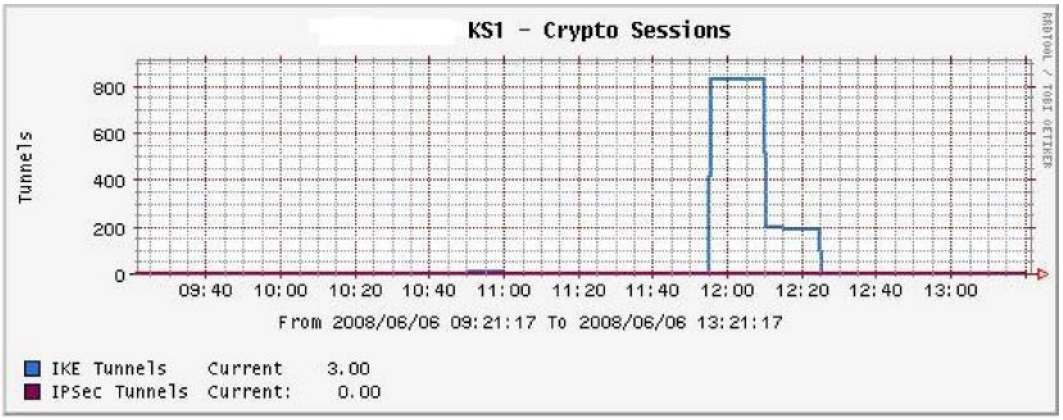
Figure 48. SNMP Polling: COOP KS Failure



5.5.1.3 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 47. 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 49. 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_KS
!
crypto isakmp policy 10
 encr aes 256
 group 14
 authentication pre-share
crypto isakmp key cisco address 192.168.1.9
crypto isakmp key cisco address 192.168.1.13
crypto isakmp key cisco address 172.18.5.2
!
crypto ipsec transform-set mygdoi-trans esp-aes esp-sha256-hmac
!
crypto ipsec profile gdoi-profile-getvpn
 set security-association lifetime seconds 7200
 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 172.16.4.2
  redundancy
   local priority 100
   peer address ipv4 172.18.5.2
!
interface FastEthernet0/1
 description Outside Interface to PE
 ip address 172.16.4.2 255.255.0.0
 ip nbar protocol-discovery
 ip flow ingress
 load-interval 30
 duplex auto
 speed auto
!
ip route 0.0.0.0 0.0.0.0 172.16.1.1
!
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_KS
!
crypto isakmp policy 10
 encr aes 256
 group 14
 authentication pre-share
crypto isakmp key cisco address 172.16.4.2
crypto isakmp key cisco address 192.168.1.9
crypto isakmp key cisco address 192.168.1.13
!
crypto ipsec transform-set mygdoi-trans esp-aes esp-sha256-hmac
!
crypto ipsec profile gdoi-profile-getvpn
```

```
 set security-association lifetime seconds 7200
 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 172.18.5.2
  redundancy
   local priority 75
   peer address ipv4 172.16.4.2
!
interface FastEthernet0/1
 description Outside interface to PE
 ip address 172.18.5.2 255.255.0.0
 duplex auto
 speed auto
 media-type rj45
 no negotiation auto
!
ip route 0.0.0.0 0.0.0.0 172.18.1.1
!
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 256
 group 14
 authentication pre-share
crypto isakmp key cisco address 172.16.4.2
crypto isakmp key cisco address 172.18.5.2
!
crypto gdoi group getvpn
 identity number 1234
 server address ipv4 172.16.4.2
 server address ipv4 172.18.5.2
!
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 192.168.1.9 255.255.255.252
 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 192.168.1.8 mask 255.255.255.252
 neighbor 192.168.1.10 remote-as 1000
 no auto-summary
```

A.2 Using Public Key Infrastructure (PKI)

A.2.1 Key Server Configuration

```
!
hostname Primary_KS
!
crypto pki trustpoint GETVPN
 enrollment url http://172.16.1.2:80
 subject-name OU=GETVPN
 revocation-check none
 auto-enroll 70
 rsakeypair pkiPKS
!
crypto pki certificate chain GETVPN
 certificate 70
 certificate ca 01
!
crypto isakmp policy 10
 encr aes 256
 group 14
 authentication rsa-sig
!
crypto ipsec transform-set mygdoi-trans esp-aes esp-sha256-hmac
!
crypto ipsec profile gdoi-profile-getvpn
 set security-association lifetime seconds 7200
 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 172.16.4.2
 redundancy
  local priority 100
  peer address ipv4 172.18.5.2
!
interface FastEthernet0/1
 description Outside Interface to PE
 ip address 172.16.4.2 255.255.0.0
 ip nbar protocol-discovery
 ip flow ingress
 load-interval 30
 duplex auto
 speed auto
!
ip route 0.0.0.0 0.0.0.0 172.16.1.1
!
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_KS
!
crypto pki trustpoint GETVPN
 enrollment url http:// 172.16.1.2:80
 subject-name OU=GETVPN
```

```
 revocation-check none
 auto-enroll 70
 rsakeypair pkiCOOP
!
crypto pki certificate chain GETVPN
 certificate 6F
 certificate ca 01
!
crypto isakmp policy 10
 encr aes 256
 group 14
 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 7200
 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 172.18.5.2
 redundancy
   local priority 75
   peer address ipv4 172.16.1.2
```

```
!
interface FastEthernet0/1
 description Outside interface to PE
 ip address 172.18.5.2 255.255.0.0
 duplex auto
 speed auto
 media-type rj45
 no negotiation auto
!
ip route 0.0.0.0 0.0.0.0 172.18.1.1
!
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://172.16.1.2:80
 subject-name OU=GETVPN
 revocation-check none
 auto-enroll 70
 rsakeypair pkiGM1
!
crypto pki certificate chain GETVPN
 certificate 6E
 certificate ca 01
!
crypto isakmp policy 10
 encr aes 256
 group 14
```

```
 authentication rsa-sig
!
crypto gdoi group getvpn
 identity number 1234
 server address ipv4 172.16.1.2
 server address ipv4 172.18.5.2
!
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 10.1.1.9 255.255.255.252
 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 10.1.1.8 mask 255.255.255.252
 neighbor 10.1.1.10 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 2016
CRL NextUpdate timer: 14:19:08 est Feb 4 2012
Current primary storage dir: flash:
Database Level: Names - subject name data written as <serialnum>.cnm
```

A.4 G-IKEv2 Configuration Using PKI

A.4.1 Key Server Configuration

```
hostname Primary_KS
!
crypto pki trustpoint GETVPN
 enrollment url http://172.16.1.2:80
 subject-name OU=GETVPN
 revocation-check none
 auto-enroll 70
 rsakeypair pkiGM1
!
crypto ikev2 proposal IKE2-PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 19
!
crypto ikev2 policy IKE2-POLICY
 match fvrf any
 proposal IKE2-PROPOSAL
!
crypto ikev2 profile IKE2-PROFILE
 match identity remote email sovm@cisco.com
 identity local key-id sovm
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint GETVPN
!
crypto ipsec transform-set TEK esp-aes esp-sha-hmac
 mode tunnel
!
crypto ipsec profile IPSEC-PROFILE-G-IKEv2
 set security-association lifetime seconds 7200
 set transform-set TEK
 set ikev2-profile IKE2-PROFILE

crypto gkm group GKM-GROUP
 identity number 130
 server local
  no gdoi
  gikev2 IKE2-PROFILE
  rekey algorithm aes 256
```

```
  rekey sig-hash algorithm sha512
  rekey retransmit 40 number 3
  rekey authentication mypubkey rsa getvpn-export-general
  rekey transport unicast
  sa ipsec 1
    profile IPSEC-PROFILE-G-IKEv2
    match address ipv4 get-acl
    replay time window-size 5
    no tag
   address ipv4 10.1.10.1
   redundancy
    local priority 100
    peer address ipv4 10.1.20.1
    protocol version optimize

interface FastEthernet0/1
 description Outside Interface to PE
 ip address 10.1.10.1 255.255.0.0

access-list get-acl remark ACL Policies to be pushed to GMs
access-list get-acl deny esp any any
access-list get-acl deny tcp any any eq tacacs
access-list get-acl deny tcp any eq tacacs any
access-list get-acl deny tcp any any eq ssh
access-list get-acl deny tcp any eq ssh any
access-list get-acl deny tcp any any eq bgp
access-list get-acl deny tcp any eq bgp any
access-list get-acl deny ospf any any
access-list get-acl deny eigrp any any
access-list get-acl deny pim any 224.0.0.0 0.0.0.255
access-list get-acl deny udp any eq isakmp any eq isakmp
access-list get-acl deny udp any any eq 848
access-list get-acl permit ip 10.2.0.0 0.0.255.255 10.2.0.0 0.0.255.255
```

A.4.2 COOP Key Server Configuration

```
hostname COOP_KS
!
crypto pki trustpoint GETVPN
 enrollment url http://172.16.1.2:80
 subject-name OU=GETVPN
 revocation-check none
 auto-enroll 70
```

```
 rsakeypair pkiGM1
!
crypto ikev2 proposal IKE2-PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 19
!
crypto ikev2 policy IKE2-POLICY
 match fvrf any
 proposal IKE2-PROPOSAL
!
crypto ikev2 profile IKE2-PROFILE
 match identity remote email sovm@cisco.com
 identity local key-id sovm
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint GETVPN
!
crypto ipsec transform-set TEK esp-aes esp-sha-hmac
 mode tunnel
!
crypto ipsec profile IPSEC-PROFILE-G-IKEv2
 set security-association lifetime seconds 7200
 set transform-set TEK
 set ikev2-profile IKE2-PROFILE

crypto gkm group GKM-GROUP
 identity number 130
 server local
  no gdoi
  gikev2 IKE2-PROFILE
  rekey algorithm aes 256
  rekey sig-hash algorithm sha512
  rekey retransmit 40 number 3
  rekey authentication mypubkey rsa getvpn-export-general
  rekey transport unicast
  sa ipsec 1
    profile IPSEC-PROFILE-G-IKEv2
    match address ipv4 get-acl
    replay time window-size 5
    no tag
   address ipv4 10.1.20.1
```

```
   redundancy
    local priority 75
    peer address ipv4 10.1.20.1
    protocol version optimize

interface FastEthernet0/1
 description Outside Interface to PE
 ip address 10.1.20.1 255.255.0.0

access-list get-acl remark ACL Policies to be pushed to GMs
access-list get-acl deny esp any any
access-list get-acl deny tcp any any eq tacacs
access-list get-acl deny tcp any eq tacacs any
access-list get-acl deny tcp any any eq ssh
access-list get-acl deny tcp any eq ssh any
access-list get-acl deny tcp any any eq bgp
access-list get-acl deny tcp any eq bgp any
access-list get-acl deny ospf any any
access-list get-acl deny eigrp any any
access-list get-acl deny pim any 224.0.0.0 0.0.0.255
access-list get-acl deny udp any eq isakmp any eq isakmp
access-list get-acl deny udp any any eq 848
access-list get-acl permit ip 10.2.0.0 0.0.255.255 10.2.0.0 0.0.255.255
```

A.4.3 Group Member Configuration

```
hostname GroupMember-1
!
crypto pki trustpoint GETVPN
 enrollment url http://172.16.1.2:80
 subject-name OU=GETVPN
 revocation-check none
 auto-enroll 70
 rsakeypair pkiGM1
!
crypto ikev2 proposal IKE2-PROPOSAL
 encryption aes-cbc-256
 integrity sha256
 group 19
!
crypto ikev2 policy IKE2-POLICY
 match fvrf any
 proposal IKE2-PROPOSAL
```

```
!
crypto ikev2 profile IKE2-PROFILE
 match identity remote key-id sovm
 identity local email sovm@gmail.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint GETVPN
!
crypto gkm group GKM-GROUP
 identity number 130
 server address ipv4 10.1.10.1
 server address ipv4 10.1.20.1
 client protocol gikev2 IKE2-PROFILE
 client registration interface Ethernet0/0
!
crypto map GETVPN_MAP 10 gdoi
 set group GKM-GROUP
!
interface Ethernet0/0
 ip address 10.1.1.101 255.255.255.0
 crypto map GETVPN_MAP
```

Appendix B. Steps to upgrade Key Servers and Group Members

This section focusses on the steps to upgrade the IOS and/or IOS-XE versions on the Key Servers and Group Members in a GETVPN network. The intent of these steps is to minimize changes in the TEK and KEK during the upgrades of the Key Servers. Setting the TEK lifetime to 7200 seconds (and KEK 86400 seconds), would give a two hour window to upgrade the Key Servers where no key changes are needed.

If an upgrade of both Key Servers and Group Members is required, it is recommended to start with the Key Servers and then upgrade the Group Members.

Steps to upgrade the Key Servers

1. Upgrade a Secondary Key Server.
2. Wait for the COOP election to complete till the next Secondary Key Server is upgraded.
3. Repeat the above steps to upgrade all Secondary Key Servers in the COOP network.

The Secondary Key Servers should reboot and synchronize with the existing Primary Key Server. They should resume their previous role as Secondary Key Server.

4. Finally, upgrade the Primary Key Server. The Primary Key Server upgrade will force one of the Secondary Key Servers to be promoted to Primary status. Once the previous Primary reboots, it should assume the role of a Secondary Key Server.

It is recommended to complete the Key Server upgrades when there is plenty of time remaining in both the KEK and TEK.

Appendix C. Steps to change RSA Keys on Key Servers

While changing the RSA keys on the Key Servers, it is recommended to do so using the following steps. This will ensure a smooth transition from the old keys to the new with no traffic disruption.

STEP 1. Verify that there is sufficient time remaining with the KEK and TEK on the Key Servers. This can be checked using the command "`show crypto gdoi ks policy`". The intent is to complete the change on all the Key Servers with sufficient time remaining in both the KEK and TEK. If it is not possible to complete the RSA key change within the remaining KEK/TEK lifetime, it is recommended to wait for a rekey before starting the process (steps 2 thru 4 below).

STEP 2. Remove the RSA keys on all the Key Servers starting with the Primary KS.

a. Use the command '`show crypto gdoi ks policy`' to find the RSA key associated with the gdoi group.

b. Use the command '`crypto key zeroize rsa <key name>`' to delete the RSA key from each of the Key Servers

STEP 3. Generate an exportable RSA key using the following command on the Primary Key Server:

```
Primary-KS(config)#crypto key generate rsa modulus 2048 label <key name> exportable
```

STEP 4. Export the RSA key generated on the Primary Key Server to all Secondary Key Servers. Here is a sample output of the export and import process using the console option. Instead of console the keys can be exported to and imported from a TFTP location also.

Export the RSA key on the Primary Key Server:

```
Primary-KS(config)#crypto key export rsa <key name> pem terminal 3des <pass code>

% Key name: <key name>

Usage: General Purpose Key

Key data:

-----BEGIN PUBLIC KEY-----

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCuKbSROW7eSqxC+IjB0ipplVkT

..

NtSRSR51ooWQW5CXRwIDAQAB

-----END PUBLIC KEY-----

-----BEGIN RSA PRIVATE KEY-----

Proc-Type: 4,ENCRYPTED

DEK-Info: DES-EDE3-CBC,B2CE8D823EE52FDC

Zi82W/lX3u0WiHN0ezi6qH5Jeo1baptdqzLlVk2jioAyZabWJqc7+svFY+DJ8rT+

...

p3dHnQSBaLu1pH3YI9gebQhMgqH6Ie00ucEYVl4/jArzUjifjdCvkQ==

-----END RSA PRIVATE KEY-----
```

Import the RSA key on each of the Secondary Key Servers:

Execute the below command on the configuration prompt of the router. "*Pass code*" is same as the one used to export the key.

```
Secondary-KS(config)#crypto key import rsa <key name> pem terminal <pass code>
% Enter PEM-formatted public General Purpose key or certificate.
% End with a blank line or "quit" on a line by itself.
<Paste the public key from the output of the key export. Paste
the hexadecimel information the lines marked BEGIN and END.>
quit
% Enter PEM-formatted encrypted private General Purpose key.
% End with "quit" on a line by itself.
<Paste the private key from the output of the key export. Paste
the hexadecimel information the linesmarkedBEGIN and END.>
quit
!
```

STEP 5. Now the next scheduled rekey from the Primary Key Server will be rejected by the Group Members since they still have the old RSA keys. This is the expected behavior. Following is a syslog message displayed on the Group Member when this occurs:

```
GM1#
*Jun   5  18:20:22.383:  %CRYPTO-6-IKMP_MODE_FAILURE:  Processing  of  GDOI  mode
failed with peer at 10.0.8.1
```

STEP 6. As a result, GMs will re-register with their KSs just before the expiry of their TEK/KEK. After the registration, the Group Members should have the latest policies and keys (including the new RSA keys).

Appendix D. Recent Features and Enhancements

The following table shows the recent GETVPN features and their corresponding releases.

Feature Name	IOS Version	IOS-XE Version
GETVPN Key Server Initial	Initial Release	XE 3.9S
VRF Aware GM	15(0)1M	XE 3.12
GETVPN GM Removal & Policy Trigger	15.2(1)T	XE 3.8S
GDOI MIB Support	15.2(1)T	XE 3.8S
GETVPN IPv6 Dataplane Support	15.2(3)T	15.3(1)S/XE 3.8S
GETVPN Suite-B support	15.2(4)M	XE 3.9S
GETVPN support of IPsec Inline Tagging for Cisco TrustSec	15.3(2)T	15.3(2)S/XE3.9S
GETVPN Resiliency **Long SA Lifetime** **Periodic Reminder Sync-Up Rekey** **Pre-Positioned Rekey**	15.3(2)T	15.3(2)S/XE3.9S
GETVPN Traceability and Debugging enhancements	15.3(2)T	15.3(2)S/XE3.9S
GM Error Recovery	15.3(3)M	15.3(3)S / XE 3.10S
GETVPN Certificate Revocation List (CRL) Checking	15.3(3)M	15.3(3)S/XE 3.10S
GDOI / IKE Separation	15.4(1)T	15.4(1)S/XE 3.11S
GETVPN Routing Awareness for BGP	15.4(3)M	15.4(3)S/XE 3.13S
GETVPN GDOI Bypass	15.4(3)M	15.4(3)S/XE 3.13S
ASR9k Specific	N/A	N/A
G-IKEv2	15.5(1)T	15.5(1)S/XE 3.14S
8K GM Scale Improvements	15.5(1)T	15.5(1)S/XE 3.14S (KS only)
IP-D3P & ID-ACK for Interoperability (KS only)	15.5(1)T	15.5(1)S/XE 3.14S
IP-D3P support on GM	N/A	IOS XE Fuji 16.7.1
IKEv2 profile based GIKEv2 authorization support	N/A	IOS XE Fuji 16.8.1
GETVPN Policy Change Rekey Enhancement (ASR1k Specific)	N/A	IOS XE Fuji 16.8.1

* - Tentative release

Appendix E. Abbreviations and Acronyms

The following table lists some common abbreviations and acronyms used to discuss GETVPN. 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 subscriber line access multiplexer
FIB	Forwarding Information Base
FWSM	Firewall Switching Module (6500/7600)
GDOI	Group Domain Of Interpretation
GETVPN	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
IP*	Internet Protocol
IPsec	IP security
ISP*	Internet service provider
KEK	key encryption key
KS	key server
L2TP	Layer 2 Tunneling Protocol

Abbreviation or Acronym	Expansion
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
VTI	Virtual Tunnel Interface
WAAS	Wide Area Application Services

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **www.cisco.com/go/offices.**

Printed in USA C07-584888-04 10/18