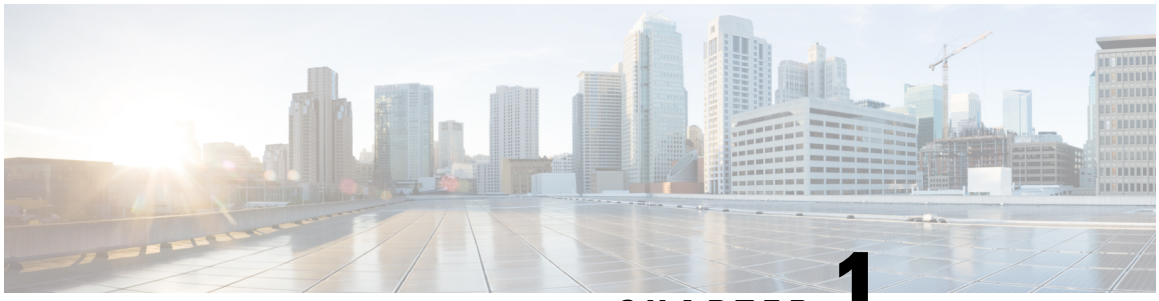
Cisco Group Encrypted Transport VPN Configuration Guide, Cisco IOS XE Fuji 16.8.x

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E for Catalyst Switching and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.

CHAPTER 2

Cisco Group Encrypted Transport VPN

Cisco Group Encrypted Transport VPN (GET VPN) is a set of features that are necessary to secure IP multicast group traffic or unicast traffic over a private WAN that originates on or flows through a Cisco IOS device. GET VPN combines the keying protocol Group Domain of Interpretation (GDOI) with IP security (IPsec) encryption to provide users with an efficient method to secure IP multicast traffic or unicast traffic. GET VPN enables the router to apply encryption to nontunneled (that is, "native") IP multicast and unicast packets and eliminates the requirement to configure tunnels to protect multicast and unicast traffic.

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

This document describes how to configure, verify, and troubleshoot Cisco GET VPN.

Cisco Group Encrypted Transport VPN provides the following benefits:

- Provides data security and transport authentication, helping to meet security compliance and internal regulation by encrypting all WAN traffic
- Enables high-scale network meshes and eliminates complex peer-to-peer key management with group encryption keys
- For Multiprotocol Label Switching (MPLS) networks, maintains network intelligence such as full-mesh connectivity, natural routing path, and quality of service (QoS)
- Grants easy membership control with a centralized key server
- Helps ensure low latency and jitter by enabling full-time, direct communications between sites, without requiring transport through a central hub
- Reduces traffic loads on customer premises equipment (CPE) and provider-edge (PE) encryption devices by using the core network for replication of multicast traffic, avoiding packet replication at each individual peer site

Finding Feature Information

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

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

Prerequisites for Cisco Group Encrypted Transport VPN

- You must be using Cisco IOS XE Release 2.3 or later.
- You should be knowledgeable about IPsec and Internet Key Exchange (IKE).
- You should know how to configure multicast and unicast routing on a Cisco IOS XE global router.
- When the IKE policy is configured, the IKE lifetime should be set to the minimum of 5 minutes so that unnecessary resources are not wasted on the maintenance of the IKE security association (SA). After the registration IKE SA is established, the registration SAs no longer have to be maintained because the rekey SA has been created and will be used to accept future rekeys.
- When the group rekey lifetime is configured with 300 seconds and forced rekey with policy change is perfomed, you might face network issues. To overcome this issue, one of the following is recommended for group rekey (KEK):
  - Set the lifetime to three times of TEK lifetime configured in transform-set.
  - Set the group rekey lifetime to default value, which is 24 hours (86400 seconds)
  - Configure rekey lifetime as 7200 seconds (2 hours)

Restrictions for Cisco Group Encrypted Transport VPN

- If you are encrypting high packet rates for counter-based antireplay, ensure that you do not make the lifetime too long or it can take several hours for the sequence number to wrap. For example, if the packet rate is 100 kilopackets per second, the lifetime should be configured as fewer than 11.93 hours so that the SA is used before the sequence number wraps.
- Cisco ASR 1000 Series Aggregation Routers with virtual-ppp interface cannot be configured as GETVPN group member.
- In Cisco IOS XE software, an inclusive port range for users to access a network cannot be matched in the extended ACL using the **permit** command.

- For unicast traffic and counter-based antireplay, the sequence numbers may be out of sync between the group members if one of the group members goes down and comes back up. For example: There is traffic from group member 1 to group member 2, and the last sequence number is *n* . Group member 1 goes down and comes back up. The sequence number of the SA at group member 1 now starts with 1, but group member 2 is expecting continuation from the previous sequence number (*n* + 1). This situation causes subsequent traffic from group member 1 to be dropped until the sequence number on group member 1 reaches *n* or the next rekey.
- When you configure transport mode traffic selectors, it is possible to have transport mode SAs. SAs occur when the packet size exceeds the MTU, and the packet cannot be forwarded.
- Transport mode should be used only for Group Encrypted Transport VPN Mode (GM) to GM traffic.
- If you are overriding the don't fragment bit (df-bit) setting in the IP header of encapsulated packets, you must configure the override commands in global configuration mode. GET VPN does not honor the interface configuration. This restriction is limited only to GET VPN. IPsec accepts both global configuration- and interface-specific override commands.
- Counter-based antireplay is not recommended and works only if there are two group members in a group.
- The GET VPN Time-Based Anti-Replay feature does not support Encapsulating Security Payload (ESP) transport mode in Cisco ASR 1000 Series Aggregation Services Routers and Cisco 4330 Integrated Services Router.
- Because Path MTU Discovery (PMTUD) does not work for GET VPN, there is a possibility that encapsulated packets could be dropped when the df-bit is set and the MTU of an intermediate link is less than the size of the encapsulated packet. In such an event, the router that drops the packet sends a notification to the source IP address on the packet, indicating that the packet has been dropped because the router could not fragment the packet due to the df-bit setting. In GET VPN, this message goes past the encapsulating endpoint directly to the source of the data due to the header preservation feature of GET VPN. Thus, the encapsulating router never knows that it has to fragment the packet to a smaller size before setting the df-bit after encapsulation. It continues to set the df-bit on the packets and they continue to be dropped at the intermediate router. (This is known as black-holing the traffic.)
- In Cisco IOS XE Release 3.5S and earlier releases, key servers cannot be configured using Cisco IOS XE images. They must be configured using Cisco IOS T-based or mainline-based images. This is not a restriction in Cisco IOS XE Release 3.6S and newer releases.
- Because of crypto engine optimization, the time-based antireplay (TBAR) overhead is 16 bytes instead of 12 bytes.
- GET VPN uses TBAR Cisco Metadata Protocol to carry TBAR information. Cisco IOS software uses 12-byte header and Cisco IOS XE uses 16-byte header. Cisco IOS XE software configured on GETVPN group members and using TBAR for anti-replay will have an effective mtu ("cleartext mtu") of the ipsec traffic as 4 bytes lower than group members that configured with Cisco IOS software. When migrating GET VPN group member from Cisco IOS software to Cisco IOS XE software, the reduction in the 4 bytes might result in unexpected performance issues.
- To ensure normal traffic flow for a GET VPN configuration on Cisco ASR 1000 Series Aggregation Services Routers, a TBAR window size greater than 20 seconds is recommended in Cisco IOS XE Release 3.12S and earlier releases, Cisco IOS XE Release 3.14S and Cisco IOS XE Release 3.15S. In Cisco IOS XE Release 3.13S, Cisco IOS XE Release 3.16S and later releases, a TBAR window size lesser than 20 seconds is permitted.

- Crypto maps are not supported on tunnel interface and port-channel interface. However, as an exception to the rule, crypto map for GDOI is supported on tunnel interfaces.
- Crypto maps are not supported on VLAN interfaces.
- RSVP as used in Mediatrace sets the "Router Alert" IP option flag. The Cavium N2 crypto accelerator does not support the use of IP options. Therefore, Mediatrace will fail with IPsec encryption on ASR1000 with Cavium N2. Mediatrace will fail with GETVPN encryption (IPSec with header preservation) on ASR1000 with Cavium N2.
- Deny statements can only be added locally to a GM. Permit statements are not supported in locally configured policies. In case of a conflict, a local policy overrides the policy downloaded from a KS.
- In Cisco ASR 1000 Series Aggregation Services Routers, when there is a failure to reregister, the outbound flow from QFP is not removed since a dummy ACE is pushed instead of a real ACE. As a result, when the SA expires, the GM will continue to encrypt outbound traffic using an expired SPI, instead of dropping the traffic locally. The traffic eventually gets dropped on the receiving GM due to an invalid SPI mechanism.
- While configuring an IPv6 access list on a Key Server, do not use the **ahp** option with the **permit** or **deny** commands.
- **SSO Restrictions**
- Cisco ASR 1000 Series Routers support stateful IPsec sessions on Embedded Services Processor (ESP) switchover. During ESP switchover, all IPsec sessions will stay up and no user intervention is needed to maintain IPsec sessions.
- For an ESP reload (no standby ESP), the SA sequence number restarts from 0. The peer router drops packets that do not have the expected sequence number. You may need to explicitly reestablish IPSec sessions to work around this issue for systems that have a single ESP after an ESP reload. Traffic disruption might happen over the IPSec sessions in such cases for the duration of the reload.
- The Cisco ASR 1000 Series Router currently does not support Stateful Switchover (SSO) IPsec sessions on Route Processors (RPs). The IPsec sessions will go down on initiation of the switchover, but will come back up when the new RP becomes active. No user intervention is needed. Traffic disruption might happen over the IPSec sessions for the duration of the switchover, until the sessions are back up.
- Cisco ASR 1000 Series Router does not support stateful ISSU for IPsec sessions. Before performing an ISSU, you must explicitly terminate all existing IPSec sessions or tunnels prior to the operation and reestablish them post ISSU. Specifically, ensure that there are no half-open or half-established IPsec tunnels present before performing ISSU. To do this, we recommend a interface shutdown in the case of interfaces that may initiate a tunnel setup, such as a routing protocol initiating a tunnel setup, or interfaces that have keepalive enabled, or where there is an auto trigger for an IPsec session. Traffic disruption over the IPsec sessions during ISSU is obvious in this case.

Information About Cisco Group Encrypted Transport VPN

Cisco Group Encrypted Transport VPN Overview

Networked applications such as voice and video increase the need for instantaneous, branch-interconnected, and QoS-enabled WANs. The distributed nature of these applications results in increased demands for scale.

At the same time, enterprise WAN technologies force businesses to trade off between QoS-enabled branch interconnectivity and transport security. As network security risks increase and regulatory compliance becomes essential, GET VPN, a next-generation WAN encryption technology, eliminates the need to compromise between network intelligence and data privacy.

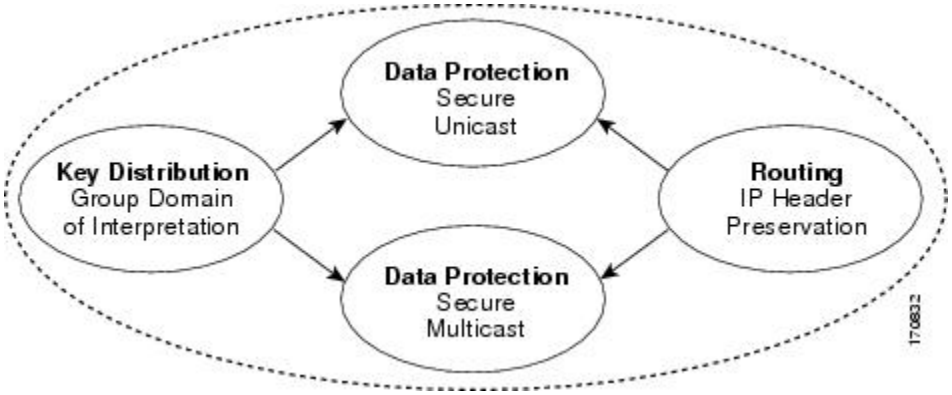
With GET, Cisco provides tunnelless VPN, which eliminates the need for tunnels. Meshed networks, by removing the need for point-to-point tunnels, can scale higher while maintaining network-intelligence features critical to voice and video quality. GET is a standards-based security model that is based on the concept of "trusted" group members. Trusted member routers use a common security methodology that is independent of any point-to-point IPsec tunnel relationship. Also, "any-any" networks, by using trusted groups instead of point-to-point tunnels, can scale higher while maintaining network-intelligence features (such as QoS, routing, and multicast), which are critical to voice and video quality.

GET-based networks can be used in a variety of WAN environments, including IP and MPLS. MPLS VPNs that use this encryption technology are highly scalable, manageable, and cost-effective, and they meet government-mandated encryption requirements. The flexible nature of GET allows security-conscious enterprises either to manage their own network security over a service provider WAN service or to offload encryption services to their providers. GET simplifies securing large Layer 2 or MPLS networks that require partial or full-mesh connectivity.

Cisco Group Encrypted Transport VPN Architecture

GET VPN encompasses Multicast Rekeying, a way to enable encryption for "native" multicast packets, and unicast rekeying over a private WAN. Multicast Rekeying and GET VPN is based on GDOI as defined in Internet Engineering Task Force (IETF) RFC 3547. In addition, there are similarities to IPsec in the area of header preservation and SA lookup. Dynamic distribution of IPsec SAs has been added, and tunnel overlay properties of IPsec have been removed. The figure below further illustrates the GET VPN concepts and relationships.

Figure 1: GET VPN Concepts and Relationships



Key Distribution Group Domain of Interpretation

GDOI

GDOI is defined as the Internet Security Association Key Management Protocol (ISAKMP) Domain of Interpretation (DOI) for group key management. In a group management model, the GDOI protocol operates between a group member and a group controller or key server (GCKS), which establishes SAs among authorized group members. The ISAKMP defines two phases of negotiation. GDOI is protected by a Phase 1 ISAKMP

security association. The Phase 2 exchange is defined in RFC 6407. The topology shown in the figure below and the corresponding explanation show how this protocol works.

Group Member

The group member registers with the key server to get the IPsec SA or SAs that are necessary to communicate with the group. The group member provides the group ID to the key server to get the respective policy and keys for this group. These keys are refreshed periodically, and before the current IPsec SAs expire, so that there is no loss of traffic.

The output of the **show crypto isakmp sa detail** command will show the security association (SA) Authentication as "rsig" because the RSA signature is used for key encryption key (KEK) rekey authentication in GET VPN.

Key Server

The responsibilities of the key server include maintaining the policy and creating and maintaining the keys for the group. When a group member registers, the key server downloads this policy and the keys to the group member. The key server also rekeys the group before existing keys expire.

Note In Cisco IOS XE Release 3.5S and earlier releases, key servers are not supported on the Cisco ASR 1000 series routers. They must be configured using Cisco IOS T-based or mainline-based images. This is not a restriction on Cisco IOS XE Release 3.6S and newer releases.

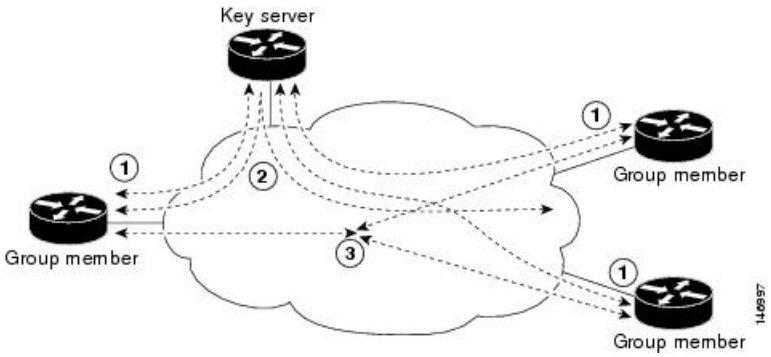
The key server has two responsibilities: servicing registration requests and sending rekeys. A group member can register at any time and receive the most current policy and keys. When a group member registers with the key server, the key server verifies the group ID that the group member is attempting to join. If this ID is a valid group ID, the key server sends the SA policy to the group member. After the group member acknowledges that it can handle the downloaded policy, the key server downloads the respective keys.

There are two types of keys that the key server can download: the key encryption key (KEK) and the traffic encryption key (TEK). The TEK becomes the IPsec SA with which the group members within the same group communicate. The KEK encrypts the rekey message.

The GDOI server sends out rekey messages if an impending IPsec SA expiration occurs or if the policy has changed on the key server (using the command-line interface [CLI]). With CSCti89255, KEK rekeys before the KEK timer expires. The group member also starts a timer and expects to receive refreshed keys before timer expiration. If they are not received, the group member initiates a jittered re-registration prior to KEK expiry. KEK is deleted when the KEK lifetime expires.

The rekey messages may also be retransmitted periodically to account for possible packet loss. Packet loss can occur because rekey messages are sent without the use of any reliable transport. If the rekey mechanism is multicast, there is no efficient feedback mechanism by which receivers can indicate that they did not receive a rekey message, so retransmission seeks to bring all receivers up to date. If the rekey mechanism is unicast, the receivers will send an acknowledgment message.

Figure 2: Protocol Flows That Are Necessary for Group Members to Participate in a Group



The topology shows the protocol flows that are necessary for group members to participate in a group, which are as follows:

1. Group members register with the key server. The key server authenticates and authorizes the group members and downloads the IPsec policy and keys that are necessary for them to encrypt and decrypt IP multicast packets.
2. As needed, the key server "pushes" a rekey message to the group members. The rekey message contains a new IPsec policy and keys to use when old IPsec SAs expire. Rekey messages are sent in advance of the SA expiration time to ensure that valid group keys are always available.
3. The group members are authenticated by the key server and communicate with other authenticated group members that are in the same group using the IPsec SAs that the group members have received from the key server.

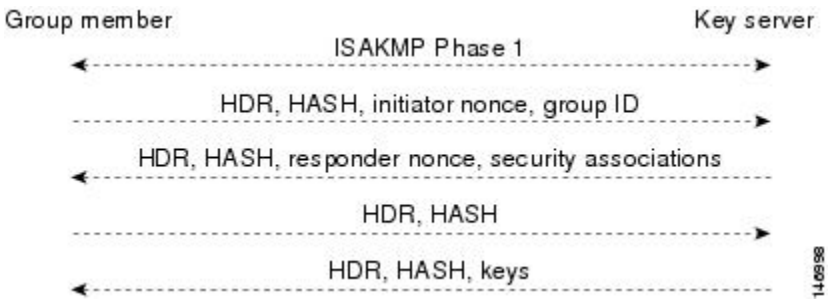
How Protocol Messages Work with Cisco Software

Multicast Rekeying uses the GDOI protocol (RFC 6407) to distribute the policy and keys for the group. The GDOI protocol is between a key server and a group member. The key server creates and maintains the policy and keys, and it downloads the policy and keys to the authenticated group members.

The GDOI protocol is protected by an ISAKMP Phase 1 exchange. The GDOI key server and the GDOI group member must have the same ISAKMP policy. This Phase 1 ISAKMP policy should be strong enough to protect the GDOI protocol that follows. The GDOI protocol is a four-message exchange that follows the Phase 1 ISAKMP policy. The Phase 1 ISAKMP exchange can occur in main mode or aggressive mode.

The figure below shows the ISAKMP Phase 1 exchange.

Figure 3: ISAKMP Phase 1 Exchange and GDOI Registration



The ISAKMP Phase 1 messages and the four GDOI protocol messages are referred to as the GDOI registration, and the entire exchange that is shown is a unicast exchange between the group member and the key server.

During the registration, if the rekey mechanism is multicast, the group member receives the address of the multicast group and registers with the multicast group that is required to receive the multicast rekeys.

The GDOI protocol uses User Datagram Protocol (UDP) port 848 (with Network Address Translation-Traversal (NAT-T), it floats to 4500).

IPsec

IPsec is a well-known RFC that defines an architecture to provide various security services for traffic at the IP layer. The components and how they fit together with each other and into the IP environment are described in IETF RFC 2401.

Communication Flow Between Key Servers and Group Members to Update IPsec SAs

Key servers and group members are the two components of the GET VPN architecture. The key server holds and supplies group authentication keys and IPsec SAs to the group members.

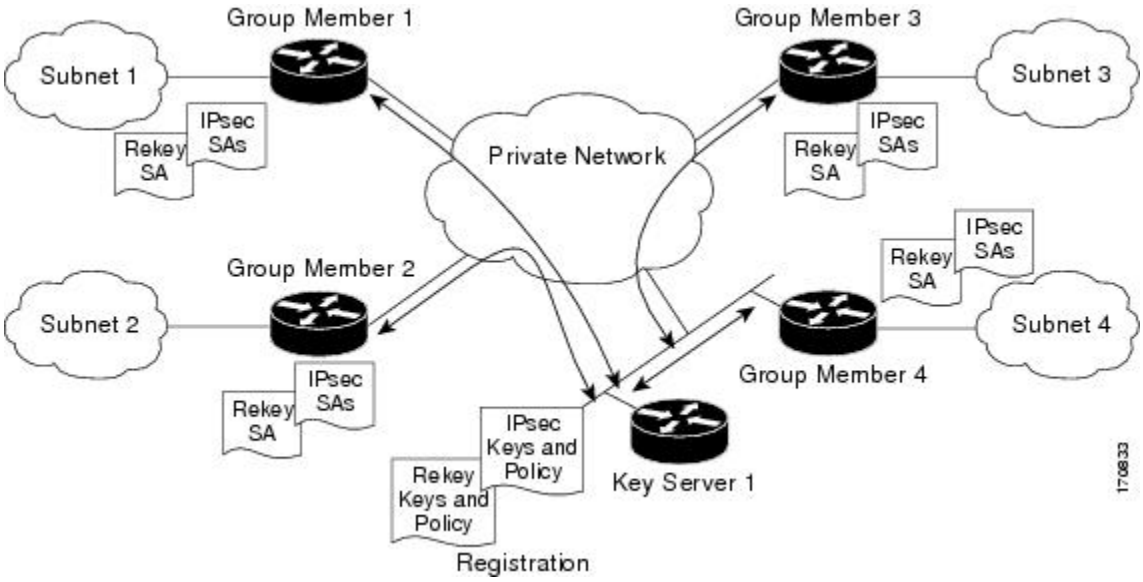
Group members provide encryption service to the interesting traffic (traffic that is worthy of being encrypted and secured by IPsec).

Communication among the key server and group members is encrypted and secured. GDOI supports the use of two keys: the TEK and the KEK. The TEK is downloaded by the key server to all the group members. The downloaded TEK is used by all the group members to communicate securely among each other. This key is essentially the group key that is shared by all the group members. The group policies and IPsec SAs are refreshed by the key server using periodic rekey messages to the group members. The KEK is also downloaded by the key server and is used by the group members to decrypt the incoming rekey messages from the key server.

The key server generates the group policy and IPsec SAs for the GDOI group. The information generated by the key server includes multiple TEK attributes, traffic encryption policy, lifetime, source and destination, a Security Parameter Index (SPI) ID that is associated with each TEK, and the rekey policy (one KEK).

The figure below illustrates the communication flow between group members and the key server. The key server, after receiving registration messages from a group member, generates the information that contains the group policy and new IPsec SAs. The new IPsec SA is then downloaded to the group member. The key server maintains a table that contains the IP address of each group member per group. When a group member registers, the key server adds its IP address in its associated group table, thus allowing the key server to monitor an active group member. A key server can support multiple groups. A group member can be part of multiple groups.

Figure 4: Communication Flow Between Group Members and the Key Server



IPsec and ISAKMP Timers

IPsec and ISAKMP SAs are maintained by the following timers:

- TEK lifetime-Determines the lifetime of the IPsec SA. Before the end of the TEK lifetime, the key server sends a rekey message, which includes a new TEK encryption key and transforms as well as the existing KEK encryption keys and transforms. The TEK lifetime is configured only on the key server, and the lifetime is "pushed down" to the group members using the GDOI protocol. The TEK lifetime value depends on the security policy of the network. If the **set security-association lifetime** command is not configured, the default value of 86,400 seconds takes effect. To configure a TEK lifetime, see the "Configuring an IPsec Lifetime Timer" section.
- KEK lifetime-Determines the lifetime of the GET VPN rekey SAs. Before the end of the lifetime, the key server sends a rekey message, which includes a new KEK encryption key and transforms and new TEK encryption keys and transforms. The KEK lifetime is configured only on the key server, and the lifetime is pushed down to group members dynamically using the GDOI protocol. The KEK lifetime value should be greater than the TEK lifetime value (it is recommended that the KEK lifetime value be at least three times greater than the TEK lifetime value). If the **rekey lifetime** command is not configured, the default value of 86,400 seconds takes effect. To configure a KEK lifetime, see the "Configuring a Multicast Rekey" section.

Note By default, the KEK lifetime is 86,400 seconds. From Cisco IOS XE Everest 16.6, a KEK lifetime of 86,400 seconds or longer is considered a long SA lifetime, and the rekey behavior is as per the long SA lifetime funtionality described in the chapter *GET VPN Resiliency*.

If you do not want the KEK lifetime to be a long SA lifetime, configure a lifetime less than 86,400 seconds.

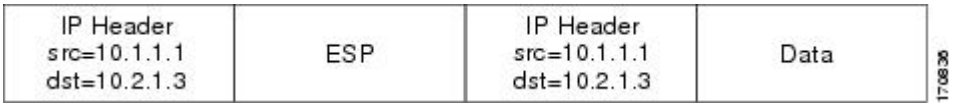
- ISAKMP SA lifetime-Defines how long each ISAKMP SA should exist before it expires. The ISAKMP SA lifetime is configured on a group member and on the key server. If the group members and key servers do not have a cooperative key server, the ISAKMP SA is not used after the group member registration. In this case (no cooperative key server), the ISAKMP SA can have a short lifetime (a minimum of 60 seconds). If there is a cooperative key server, all key servers must have long lifetimes to keep the ISAKMP SA "up" for cooperative key server communications. If the **lifetime** command is not configured, the default value of 86,400 seconds takes effect. To configure an ISAKMP SA lifetime, see the “Configuring an ISAKMP Lifetime Timer” section.

Address Preservation

The following section describes address preservation in GET VPN.

As shown in the figure below, IPsec-protected data packets carry the original source and destination in the outer IP header rather than replacing them with tunnel endpoint addresses. This technique is known as IPsec Tunnel Mode with Address Preservation.

Figure 5: Header Preservation



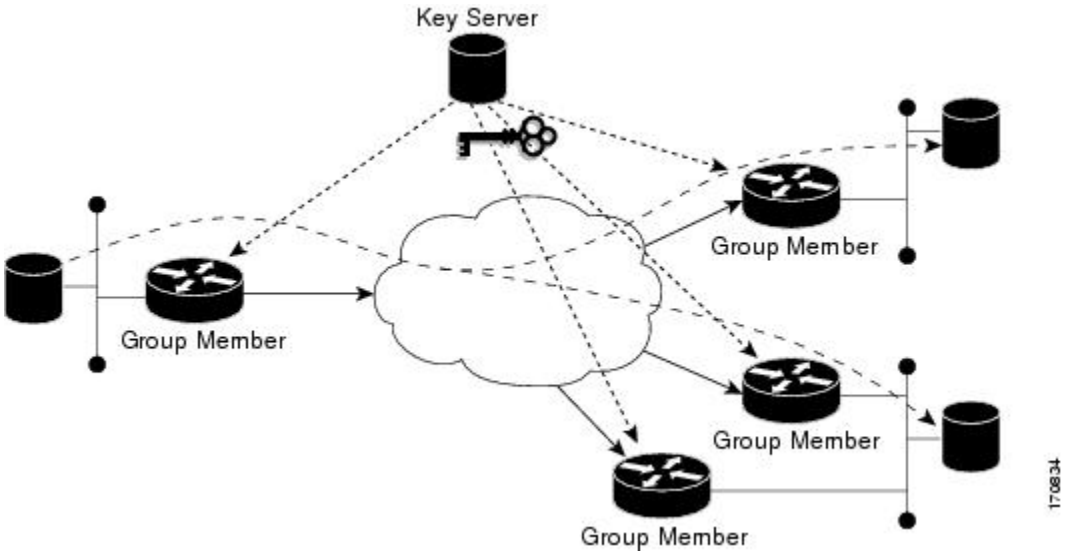
Address preservation allows GET VPN to use the routing functionality present within the core network. Address preservation allows routing to deliver the packets to any customer-edge (CE) device in the network that advertises a route to the destination address. Any source and destination matching the policy for the group will be treated in a similar manner. In the situation where a link between IPsec peers is not available, address preservation also helps combat traffic “black-hole” situations.

Header preservation also maintains routing continuity throughout the enterprise address space and in the WAN. As a result, end host addresses of the campus are exposed in the WAN (for MPLS, this applies to the edge of the WAN). For this reason, GET VPN is applicable only when the WAN network acts as a “private” network (for example, in an MPLS network).

Secure Data Plane Multicast

The multicast sender uses the TEK that is obtained from the key server and encrypts the multicast data packet with header preservation before it switches out the packet. The replication of the multicast packet is carried out in the core on the basis of the (S, G) state that is retained in the multicast data packet. This process is illustrated in the figure below.

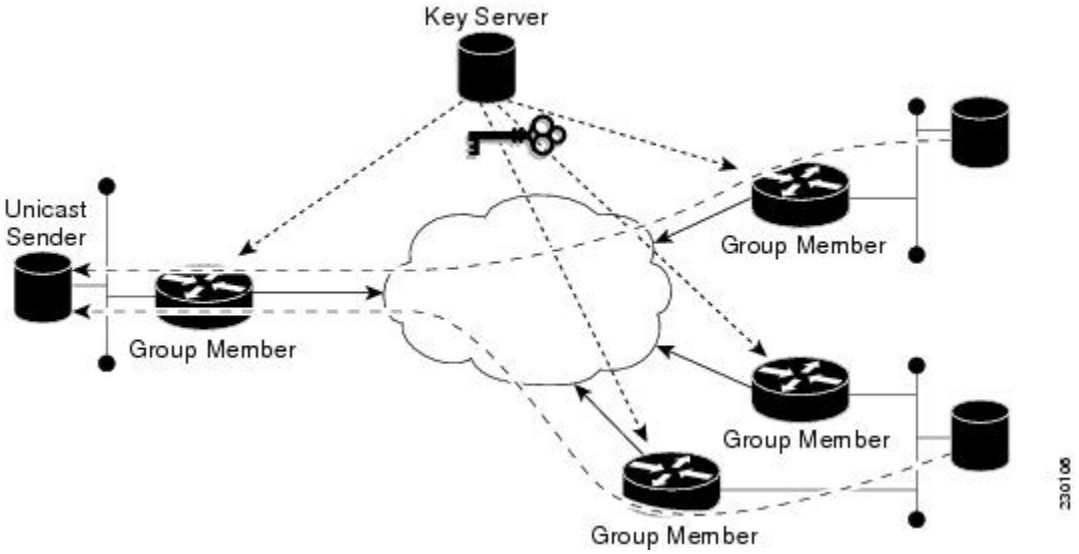
Figure 6: Secure Data Plane Multicast Process



Secure Data Plane Unicast

The unicast sender uses the TEK that is obtained from the key server and encrypts the unicast data packet with header preservation before it switches out the packet to the destination. This process is illustrated in the figure below.

Figure 7: Secure Data Plane Unicast Process



Cisco Group Encrypted Transport VPN Features

Rekeying

Rekey messages are used to refresh IPsec SAs. When the IPsec SAs or the rekey SAs are about to expire, one single rekey message for a particular group is generated on the key server. No new IKE sessions are created for the rekey message distribution. The rekey messages are distributed by the key server over an existing IKE SA.

Rekeying can use multicast or unicast messages. GET VPN supports both unicast and multicast rekeying.

With CSCti89255, KEK rekeys before the KEK timer expires. The group member also starts a timer and expects to receive refreshed keys before timer expiration. If they are not received, the group member initiates a jittered re-registration prior to KEK expiry. KEK is deleted when the KEK lifetime expires. This ensure the following:

- A safer KEK expiry checking mechanism
- A safer KEK re-registration mechanism
- Avoids use of KEK beyond configured lifetime

The following subsections give detailed rekeying information:

Rekey Sequence-Number Check

The rekey sequence-number check between the key server and the group member is conducted as follows:

1. Antireplay in GROUPKEY-PUSH messages is restored as specified in RFC 6407.
   - The group member drops any rekey message that has a sequence number lower than or equal to that of the last received rekey message.
   - The group member accepts any rekey message that has a sequence number higher than that of the last received rekey message, no matter how large the difference.
2. The sequence number is reset to 1 at the first rekey message after the KEK rekey, not at the KEK rekey message itself.

Multicast Rekeying

Multicast rekeys are sent out using an efficient multicast rekey. Following a successful registration, the group member registers with a particular multicast group. All the group members that are registered to the group receives this multicast rekey. Multicast rekeys are sent out periodically on the basis of the configured lifetime on the key server. Multicast rekeys are also sent out if the IPsec or rekey policy is changed on the key server. Triggered by the configuration change, the rekey sends out the new updated policy to all the group members with an efficient multicast rekey.

The key server pushes the rekey time back as follows:

1. If the TEK timeout is 300 seconds:

tek_rekey_offset = 90 (because 300 < 900)

If retransmissions are configured, the rekey timer is moved back more.

For three retransmissions every 10 seconds: 3 x 10

So the rekey will actually happen at (300 - 90 - 30) = 180 seconds

2. If the TEK timeout is 3600 seconds:

tek_rekey_offset = 3600 x 10 percent = 360 seconds

If retransmissions are configured, the rekey timer is moved back more.

For three retransmissions every 10 seconds: 3 x 10

So the rekey will actually happen at (3600 - 360 - 30) = 3210 seconds

When a KEK expires and when the transport mode is multicast, a multicast KEK rekey is sent. When a multicast KEK rekey is sent, the group member replaces the old KEK with the new KEK. Because it is a multicast rekey, and the retransmissions are sent, the old KEK continues to be used for encryption. This situation occurs because the group member does not receive the new KEK rekey. Hence the group member that received the multicast KEK rekey does not have the old KEK, and hence it drops these retransmissions.

The group member that did not initially receive the KEK key now receives the KEK retransmission and replaces the old KEK with the new KEK and will drop the retransmissions that will follow. For example, if five retransmissions are configured and a multicast KEK rekey with sequence number 1 is received at group member 1, all the other retransmissions with sequence numbers 2 3 4 5 6 will be dropped at the group member because the group member does not have the old KEK.

If group member 2 does not get the KEK rekey with sequence number 1 and it receives the retransmission with sequence number 2, it will drop the other retransmissions 3, 4, 5, 6.

Configuration Requirements for Multicast Rekeying

When a group member registers to a key server, it installs the KEK SA into its database. When the rekey transport is multicast the group member will use IGMP to join the multicast stream defined by the key server. The IGMP join is transmitted from the interface that contains the crypto map.

Note The IGMP traffic should be excluded from encryption via either the ACL defined on the key server or a local deny ACL on the group member.

When the key server is not reachable via the same interface as the one configured with the crypto map, it will have to manually join the stream.

Unicast Rekeying and SAs

In a large unicast group, to alleviate latency issues, the key server generates rekey messages for only a small number of group members at a time. The key server is ensured that all group members receive the same rekey messages for the new SA before the expiration of the old SA. Also, in a unicast group, after receiving the rekey message from the key server, a group member sends an encrypted acknowledge (ACK) message to the key server using the keys that were received as part of the rekey message. When the key server receives this ACK message, it notes this receipt in its associated group table, which accomplishes the following:

- The key server keeps a current list of active group members.
- The key server sends rekey messages only to active members.

In addition, in a unicast group, the key server removes the group member from its active list and stops sending the rekey messages to that particular group member if the key server does not receive an ACK message for three consecutive rekeys. If no ACK message is received for three consecutive rekeys, the group member has to fully re-register with the key server after its current SA expires if the group member is still interested in receiving the rekey messages. The ejection of a nonresponsive group member is accomplished only when the

key server is operating in the unicast rekey mode. The key server does not eject group members in the multicast rekey mode because group members cannot send ACK messages in that mode.

As in multicast rekeying, if retransmission is configured, each rekey will be retransmitted the configured number of times.

Rekey transport modes and authentication can be configured under a GDOI group.

If unicast rekey transport mode is not defined, multicast is applied by default.

If the TEK rekey is not received, the group member re-registers with the key server 60 seconds before the current IPsec SA expires. The key server has to send out the rekey before the group member re-registration occurs. If no retransmission is configured, the key server sends the rekey tek_rekey_offset before the SA expires. The tek_rekey_offset is calculated based on the configured rekey lifetime. If the TEK rekey lifetime is less than 900 seconds, the tek_rekey_offset is set to 90 seconds. If the TEK rekey lifetime is configured as more than 900 seconds, the tek_rekey_offset = (configured TEK rekey lifetime)/10. If retransmission is configured, the rekey occurs earlier than the tek_rekey_offset to let the last retransmission be sent 90 seconds before the SA expires.

The key server uses the formula in the following example to calculate when to start sending the rekey to all unicast group members. The unicast rekey process on the key server sends rekeys to unicast group members in groups of 50 within a loop. The time spent within this loop is estimated to be 5 seconds.

A key server rekeys group members in groups of 50, which equals two loops. For example, for 100 group members:

Number of rekey loops = (100 group members)/50 = 2 loops:

- Time required to rekey one loop (estimation) = 5 seconds
- Time to rekey 100 group members in two loops of 50: 2 x 5 seconds = 10 seconds

So the key server pushes the rekey time back as follows:

- If the TEK timeout is 300: 300 - 10 = 290

But the start has to be earlier than the TEK expiry (as in the multicast case):

- Because 300 < 900, tek_rekey_offset = 90
- So 90 seconds is subtracted from the actual TEK time: 290 - tek_rekey_offset = 200 seconds

If retransmissions are configured, the rekey timer is moved back more:

- For three retransmissions every 10 seconds: 200 - (3 x 10) = 170
- If the TEK timeout is 3600 seconds: 3600 - 10 = 3590

But the start has to be earlier than the TEK expiry (as in the multicast case):

- Because 3600 > 900, tek_rekey_offset = 3600 x 10 percent = 360
- So 360 seconds is subtracted from the actual TEK time: 3590 - tek_rekey_offset = 3230 seconds

If retransmissions are configured, the rekey timer is moved back more:

- For three retransmissions every 10 seconds: 3230 - (3 x 10) = 3200 seconds

The tek_rekey_offset formula applies to unicast and multicast rekeying.

Rekey Behavior After Policy Changes

The table below provides a list of rekey behavior based on the security policy changes.

Table 1: Rekey Behavior After Security Policy Changes

Policy Changes	Rekey Sent?	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 transformset	Yes	The SAs of the old transform set remain active until its lifetime expires.
TEK: IPSEC profile	Yes	The SAs of the old profile remain active until its lifetime expires.
TEK:matching ACL	Yes	Outbound packet classification will use the new access control list (ACL) immediately. The old SAs are still kept 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	No	The SA with a new replay counter will be 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 receive-only	Yes	Receive-only mode is activated immediately after rekey.
TEK:disable receive-only	Yes	Receive-only mode is deactivated immediately after rekey.
KEK:SA lifetimebehavior	No	Change is applied with the next rekey.
KEK:change authentication key	Yes	Change is applied with the next rekey.
KEK:changing crypto algorithm	Yes	Change is applied immediately.

Enter the following commands for the policy changes to take effect immediately:

- Use the **clear crypto gdoi** [**group**] command on the key server.
- Use the **clear crypto gdoi** [**group**] command on all the group members.

Note The key server sends rekeys for policy updates after the administrator exits configuration mode, ensuring that the rekeys are sent when appropriate.

Note Passive-mode behavior before changing to bidirectional mode on a group member is as follows:

If an you change the SA mode on the key server to "no sa receive-only," and exits configuration mode, the rekey is sent to the group member, and you can see the state on the group member changing from "inbound only" to "inbound optional;" the state will change to "both" after an interval set by a built-in timer; about five minutes.

The key server shows this state as "both" immediately; this is done by design because all group members might be in the process of being updated.

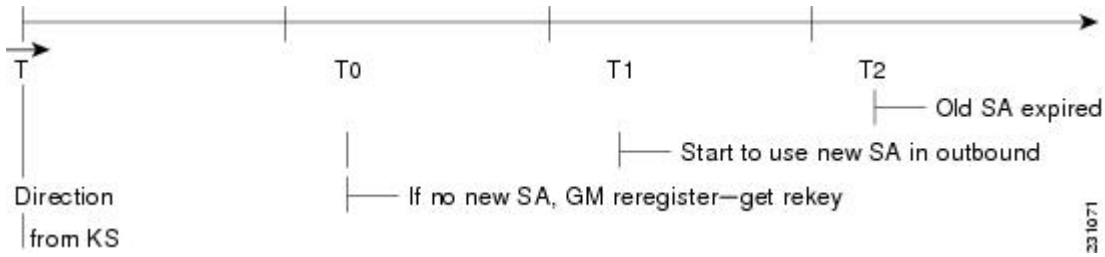
IPsec SA Usage on the Group Members

When a rekey is received and processed on a group member, the new IPsec SA (the SPI) is installed. There is a period of time when the old and the new IPsec SAs are used. After a certain specified interval, the old IPsec SA is deleted. This overlap ensures that all group members receive the current rekey and insert the new IPsec SAs. This behavior is independent of the transport method (multicast or unicast rekey transport) for the rekeys from the key server.

Approximately 30 seconds before the old SA expires, the group member starts to use the new SA in the outbound direction to encrypt the packet. Approximately 60 seconds before the old SA expires, if no new SA is received on the group member side via a rekey from the key server, the group member reregisters.

In the figure below, time T2 is when the old SA expires. T1 is 30 seconds before T2, which is when the group member (GM) starts to use the new SA in the outbound direction. T0 is another 30 seconds before T2. If no new SA is received at T0, the group member has to reregister. T is another 30 seconds from T0. The key server should send a rekey at T.

Figure 8: IPsec SA Usage on a Group Member



Configuration Changes Can Trigger a Rekey By a Key Server

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

Configuration changes on a key server can trigger a rekey by the key server. Please refer to the following sample configuration as you read through the changes that will or will not cause a rekey that are described following the example.

```
crypto ipsec transform-set gdoi-p esp-aes esp-sha-hmac
!
```

```
crypto ipsec profile gdoi-p
 set security-association lifetime seconds 900
set transform-set gdoi-p
!
crypto gdoi group diffint
 identity number 3333
 server local
  rekey algorithm aes 128
  rekey address ipv4 121
  rekey lifetime seconds 3600
  no rekey retransmit
  rekey authentication mypubkey rsa mykeys
  sa ipsec 1
   profile gdoi-p
   match address ipv4 120
   replay counter window-size 3
```

The following configuration changes on the key server will trigger a rekey from the key server:

- Any change in the TEK configuration ("sa ipsec 1" in the example):
  - If the ACL ("match address ipv4 120" in the above example) is changed. Any addition, deletion, or change in the ACL causes a rekey.
  - If TEK replay is enabled or disabled on the key server, rekey is sent.
  - Removal or addition of the IPsec profile in the TEK ("profile gdoi-p" in the example).
  - Changing from multicast to unicast transport.
  - Changing from unicast to multicast transport.

The following configuration changes on the key server will not trigger a rekey from the key server:

- Replay counter window size is changed under the TEK ("sa ipsec 1" in the example).
- Configuring or removing rekey retransmit.
- Removing or configuring the rekey ACL.
- Changing the TEK lifetime ("set security-association lifetime seconds 300" in the example) or changing the KEK lifetime ("rekey lifetime seconds 500" in the example).
- Adding, deleting, or changing the rekey algorithm ("rekey algorithm aes 128" in the example).

Commands That Trigger a Rekey

The table below is a comprehensive list of GET VPN command changes, and it shows which commands will or will not trigger a rekey. Commands are broken out based on the configuration mode in which they are entered. The table also shows when the commands take effect, regardless of whether they trigger a rekey.

Note When the KEK lifetime is changed in the GDOI group, the changes take place only when the current KEK expires and a new one is generated. You can force the changes to take place, by issuing the rekey command, **crypto gdoi ks rekey**, on the key server.

Table 2: Commands That Trigger a Rekey

Description	Command	Rekey Triggered	When Triggered	When Change Takes Effect
Mode = (config)	**configure terminal**	—	—	—
Change/delete ACL used in GDOI group (example: **rekey address ipv4** *access-list-number*[*options*])	[**no**] **access-list** *access-list-number*[*options*]	No	—	Immediately
Change/delete ACL used in IPsec profile (example: **match address ipv4** *access-list-id* | *name*[*options*])	[**no**] **access-list** *access-list-number*[*options*]	Yes	End configuration mode	**show running-config** command output on key server indicates that the policy is incomplete, the packet is still encrypted/decrypted by the existing SA, downloaded ACLs are cleared but multidimensional-tree entries are still present (by displaying **show crypto ruleset** command output), and no new SAs are downloaded and old SAs are still active in encrypt/decrypt.
Add/remove ISAKMP preshared key (arbitrary key)	**crypto isakmp key address** *peer-address*	No	—	Immediately
Add/remove ISAKMP preshared key (group member key)	**crypto isakmp key address** *peer-address*	No	—	After key encryption key (KEK) SA expires (re-registration)
Add IPsec profile	**crypto ipsec profile**	No	—	Immediately
Add/remove ISAKMP policy	**crypto isakmp policy** *priority*	No	—	Immediately
Mode = (ipsec-profile)	**crypto ipsec profile** *name*	—	—	—
Change SA lifetime (in IPsec profile)	**set security-association lifetime** *seconds*	No	—	Next rekey
Change transform-set	**set transform-set** *transform-set-name*	Yes	End configuration mode	The SAs of the old transform set remain active until the lifetime expires.

Description	Command	Rekey Triggered	When Triggered	When Change Takes Effect
Mode = (config-gdoi-group)	**crypto gdoi group** *group-name*	—	—	—
Change identity number	**identity number** *number*	No	—	Must immediately configure on the group member. The other group members keep using the TEKs and KEKs of the old group ID.
Mode = (gdoi-local-server)	**server local**	—	—	—
Change from unicast to multicast transport	**rekey transport unicast**	Yes	Immediately	After triggered rekey
Change from multicast to unicast transport	[**no**] **rekey transport unicast**	Yes	End configuration mode	After triggered rekey
Change rekey address	**rekey address ipv4** {*access-list-number* | *access-list-name*}	Yes	End configuration mode	After triggered rekey (however, changing the ACL itself will not trigger a multicast rekey)
Change rekey lifetime	**rekey lifetime seconds** *number-of-seconds*	No	—	Next rekey, but lifetime starts decrementing when the command is issued (the current lifetime is sent out with the rekey).
Enable/disable rekey retransmit	**rekey retransmit** *number-of-seconds* [**number** *number-of-retransmissions*]	No	—	Next rekey
Enable rekey authentication	**rekey authentication mypubkey rsa** *key-name*	Yes	End configuration mode	After triggered rekey
Disable rekey authentication	[**no**] **rekey authentication**	No	—	Immediately
Change rekey authentication key	**rekey authentication mypubkey rsa** *key-name*	Yes	End configuration mode	After triggered rekey
Change rekey encryption	**rekey algorithm** *type-of-encryption-algorithm*	Yes	End configuration mode	New algorithm takes effect immediately.
Mode = (gdoi-sa-ipsec)	**sa ipsec** *sequence-number*	—	—	—

Description	Command	Rekey Triggered	When Triggered	When Change Takes Effect
Change profile	**profile** *ipsec-profile-name*	Yes	End configuration mode	SAs of the old profile are still in effect until the lifetime expires.
Change ACL match	**match address** [options]	Yes	End configuration mode	After triggered rekey
Enable counter replay	**replay counter window-size** *seconds*	Yes	End configuration mode	Old SA without counter replay is still inactive until the lifetime expires.
Change replay counter value	**replay counter window-size** *seconds*	No	—	Next rekey
Enable time-based antireplay	**replay time window-size** *seconds*	Yes	End configuration mode	New SA with time-based antireplay enabled is sent, but the old SA with time-based antireplay disabled is still active until the lifetime expires.
Change time-based antireplay window	**replay time window-size** *seconds*	No	—	New time-based antireplay window is effective only after entering the **clear crypto gdoi** command on both the key server and group member.
Mode = (gdoi-coop-ks-config)	**redundancy**	—	—	—
Enable redundancy	**redundancy**	No	—	Must immediately configure on other key servers
Change local priority	**local priority** *number*	No	—	Immediately but does not force key server election
Add/remove peer address	[**no**] **peer address ipv4** *ip-address*	No	—	Next cooperative (COOP) message
Disable redundancy	[**no**] **redundancy**	No	—	Must immediately configure on other key servers

When a timeout is caused by a pseudotime synchronization, the key server checks if either the KEK or the TEK timer is scheduled to expire in next 60 seconds, and if so, combines that timeout with the pseudotime

synchronization timeout. That is, the rekey acts as both a TEK or KEK rekey and a pseudotime synchronization timeout rekey. See the "Time-Based Antireplay" section for more information on pseudotime synchronization.

Retransmitting a Rekey

Multicast rekeys are retransmitted by default. For unicast rekeys, if the key server does not receive the ACK, it retransmits the rekey. In either case, before retransmitting a rekey, the key server checks if there is a TEK or KEK rekey scheduled in the next 120 seconds. If so, it stops the current retransmission and waits for the scheduled rekey to happen.

Group Member Access Control List

For GET VPN, the traffic that has to be protected is defined statically on the key server using the ACL. The group member gets information about what has to be protected from the key server. This structure allows the key server to choose and change the policy dynamically as needed. In Secure Multicast, the key server ACL is defined inclusively. The ACL includes only the exact traffic that should be encrypted, with an implicit deny causing all other traffic to be allowed in the clear (that is, if there is no permit, all other traffic is allowed).

GET VPN employs a different philosophy: The definition of which packets should be encrypted is delivered independently. GET VPN supports only statically defined traffic selectors. Policy can be defined by using both deny and permit ACLs on the key server. Only the deny ACL is allowed to be manually configured on a group member. The policies that are downloaded from the key server and configured on the group member are merged. Any ACL that is configured on the group member has predominance over what is downloaded from the key server.

After the group member gets the ACL from the key server, the group member creates a temporary ACL and inserts it into the database. This ACL will be deleted if the group member is removed from the GDOI group for any reason. The packets that are going out of the interface are dropped by the group member if a packet matches the ACL but no IPsec SA exists for that packet.

The key server can send a set of traffic selectors, which may not exactly match the group member ACL on the group member. If such differences occur, the differences have to be merged and resolved. Because the group member is more aware of its topology than the key server, the downloaded ACLs are appended to the group member ACL. The group member ACL (except the implicit deny) is inserted into the database first, followed by the downloaded key server ACL. The database is prioritized, and the database search stops whenever a matched entry is found.

Note

- On a Key Server (KS) running Cisco IOS XE Fuji 16.8.1 or later, do not configure a deny statement as the last entry of a KS GETVPN ACL. Such a configuration is not supported. The KS ignores the last deny statement and does not include it in the KS GETVPN ACL sent to Group Members (GMs).
- On a KS running a Cisco IOS XE release earlier than Cisco IOS XE Fuji 16.8.1, configuration of a deny statement as the last entry in a KS GETVPN ACL is supported only if none of the GMs is running Cisco IOS XE Fuji 16.8.1 or later. If a GM is running Cisco IOS XE Fuji 16.8.1 or later, and the last entry in a KS GETVPN ACL is a deny statement, the encryption policy on the GM is corrupted after a rekey and the GM behaves in an undefined manner. To avoid any adverse impact due to this undefined GM behavior, do not configure a deny statement as the last entry in a KS GETVPN ACL.

If the KS or GMs in a GETVPN deployment are running Cisco IOS XE Fuji 16.8.1 or later, we recommend that you configure a permit statement as the last entry in a KS GETVPN ACL.

For information about configuring a group member ACL, see the "Configuring Group Member ACLs" section.

Behavior of a Group Member When Security Policy Changes

The behavior of a group member changes when ACL changes or any other policy changes are made in the key server. The effect of different policy changes on the behavior of the group members is explained in the following three scenarios.

Scenario 1

In the following example, the ACL has been initially configured to permit host A and host B.

```
ip access-list extended get-acl
permit ip host A host B
permit ip host B host A
```

Then the ACL is changed to permit host C and host D in the key server:

```
ip access-list extended get-acl
permit ip host C host D
permit ip host D host C
```

ACL changes affect the behavior of the group member in the following ways:

- Key server sends out a rekey to all group members immediately.
- Group member sends traffic between host A and host B in clear text immediately after rekey.
- Group member sends traffic between host C and host D in encrypted text immediately after rekey.

Note GETVPN group members of Cisco ASR 1000 Series Aggregation Services Routers and Cisco ISR G2 routers behave differently after a rekey (either triggered or periodic) that follows a ACL change or any other policy change in the key server. The group members of Cisco ISR G2 routers install the new policy without a full reregistration, while the group members of Cisco ASR 1000 Series Aggregation Services Routers will reregister to get the updated policy.

Scenario 2

The behavior of a group member changes when policy updates and transform set and time-based antireplay (TBAR) changes are made to the key server.

In this scenario, it is assumed that:

- The transform set has been changed from ESP-3DES to ESP-AES.
- The policy change occurs at 1000 seconds before the current TEK lifetime expires.

These policy changes affect the behavior of the group member in the following ways:

- The key server sends out a rekey of both old SAs (3DES) and new SAs (AES).
- Group member continues to use the old SA (3DES) for 1000 seconds until it expires.
- After the old SA expires, the group member automatically switches over to new SAs (AES).

Scenario 3

The behavior of a group member changes when other policy updates in the key server involve both ACL changes and other changes like the transform set or TBAR.

In this scenario it is assumed that:

- The ACL has been updated as specified in Scenario 1.
- The transform set was changed from ESP-3DES to ESP-AES.
- The policy change occurs 1000 seconds before the current TEK lifetime expires.

ACL changes and other policy updates affect the behavior of the group member in the following ways:

- The key server sends out a rekey that consists of both old SAs (3DES) and new SAs (AES).
- The group member sends traffic between host A and host B in clear text immediately after rekey.
- The group member sends encrypted traffic between host C and host D using old SAs (3DES) for 1000 seconds until its TEK lifetime expires.
- When old SAs (3DES) expire, the group member automatically switches to new SAs to encrypt traffic between host C and host D in AES.

Enhancement in Group Members Running Cisco IOS XE Software

Effective with Cisco IOS XE Fuji 16.8.1, the GETVPN Policy-Change Enhancement for XE-based Group Members feature enhances group members, running Cisco IOS XE software, handle policy change rekeys that require flow relocation. As a result of this feature, group members need not reregister and download again SAs and traffic that matches the old and new crypto policy is not leaked via clear text.

Note There are no changes either to scheduled rekeys or to policy change rekeys without flow relocation.

The limitations of this feature are as follows:

- A tiny window (approximately 5 to 10 milliseconds) of slight packet drop may happen during policy change rekey
- This enhancement does not apply to GM local policy change and it will still trigger GM reregistration
- KS cannot trigger policy change rekey if an older SA is present with lifetime of less than 30 seconds
- When an SA is deleted due to policy change rekey, the crypto statistics (encrypt and decrypt counters) may not updated accurately for about 1 second.
- GETVPN Suite B policy change rekey must ensure unique initialization vector (IV) in each packet, where IV = GM_SID + GM_SSID. GM allocates 90% of GM_SSID when installing new SAs and 10% is reserved for policy change rekey usage. When policy change rekey occurs, 90% of GM_SSID space is allocated for new TEKs received in rekey and the reserved 10% of GM_SSID space is allocated for the old-TEKs received in rekey. If a KS makes a policy change rekey before the expiry of old-TEKs received in the first policy change rekey and the GM has no reserved GM_SSID space for old-TEKs, the GM will reregister to refresh the policy or SA with new GM_SID.

Time-Based Antireplay

Antireplay is an important feature in a data encryption protocol such as IPSec (RFC 2401). Antireplay prevents a third party from eavesdropping on an IPsec conversation, stealing packets, and injecting those packets into a session at a later time. The time-based antireplay mechanism helps ensure that invalid packets are discarded by detecting the replayed packets that have already arrived at an earlier time.

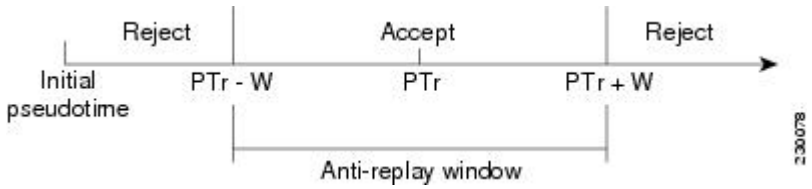
GET VPN uses the Synchronous Antireplay (SAR) mechanism to provide antireplay protection for multisender traffic. SAR is independent of real-world Network Time Protocol (NTP) clock or sequential-counter mechanisms (which guarantee packets are received and processed in order). A SAR clock advances regularly. The time tracked by this clock is called pseudotime. The pseudotime is maintained on the key server and is sent periodically to the group members within a rekey message as a time-stamp field called pseudoTimeStamp. GET VPN uses a Cisco proprietary protocol called Metadata to encapsulate the pseudoTimeStamp. Group members have to be resynchronized to the pseudotime of the key server periodically. The pseudotime of the key server starts ticking from when the first group member registers. Initially, the key server sends the current pseudotime value of the key server and window size to group members during the registration process. New attributes, such as time-based replay-enabled information, window size, and the pseudotime of the key server, is sent under the SA payload (TEK).

The group members use the pseudotime to prevent replay as follows: the pseudoTimeStamp contains the pseudotime value at which a sender created a packet. A receiver compares the pseudotime value of senders with its own pseudotime value to determine whether a packet is a replayed packet. The receiver uses a time-based antireplay "window" to accept packets that contain a time-stamp value within that window. The window size is configured on the key server and is sent to all group members.

Note You should not configure time-based antireplay if you are using a Cisco VSA as a group member.

The figure below illustrates an antireplay window in which the value PTr denotes the local pseudotime of the receiver, and W is the window size.

Figure 9: Antireplay Window



Clock Synchronization

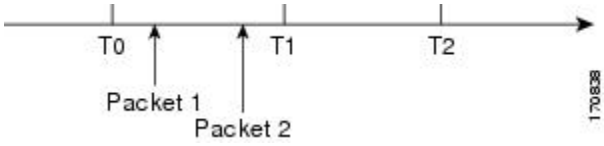
Clocks of the group members can slip and lose synchronization with the key server. To keep the clocks synchronized, a rekey message (multicast or unicast, as appropriate), including the current pseudotime value of the key server, is sent periodically, either in a rekey message or at a minimum of every 30 minutes to the group member. If a packet fails this antireplay check, the pseudotime of both the sender and receiver is printed, an error message is generated, and a count is increased.

To display antireplay statistics, use the **show crypto gdoi group** *group-name* **gm replay** command on both the sender and receiver devices. If the configuration is changed by the administrator to affect the replay method of the size configuration, the key server initiates a rekey message.

Interval Duration

A tick is the interval duration of the SAR clock. Packets sent in this duration have the same pseudoTimeStamp. The tick is also downloaded to group members, along with the pseudotime from the key server. For example, as shown in the figure below, packets sent between T0 and T1 would have the same pseudoTimeStamp T0. SAR provides loose antireplay protection. The replayed packets are accepted if they are replayed during the window. The default window size is 100 seconds. It is recommended that you keep the window size small to minimize packet replay.

Figure 10: SAR Clock Interval Duration



Antireplay Configurations

The Antireplay feature can be enabled under IPsec SA on a key server by using the following commands:

- **replay time window-size**—Enables the replay time option, which supports the nonsequential, or time-based, mode. The window size is in seconds. Use this mode only if you have more than two group members in a group.
- **replay counter window-size**—Enables sequential mode. This mode is useful if only two group members are in a group.
- **no replay counter window-size**—Disables antireplay.

Control-Plane Time-Based Antireplay

Rekey Pseudotime Check

The rekey pseudotime check between key servers and group members is conducted as follows:

- The group member calculates the allowable pseudotime difference between the key server and its own as the lesser of the configured TBAR window size, that is, the value that was configured for it in the data plane, or 30 seconds.
- The group member accepts any rekey with a pseudotime larger than its own and updates its own pseudotime to the larger value. If the difference is larger than the calculated allowable pseudotime difference, it also generates the following syslog message:

```
*Jul 28 22:56:37.503: %GDOI-3-PSEUDO_TIME_LARGE: Pseudotime difference between key server
(20008 sec) and GM (10057 sec) is larger than expected in group GET. Adjust to new pseudotime
```

- If the group member receives a rekey with a pseudotime smaller than its own but within the allowable difference, the group member accepts the rekey and updates its pseudotime value to the rekey pseudotime value.
- If the group member receives a rekey with a pseudotime smaller than its own but exceeding the allowable difference, the group member drops the rekey message and generates the following syslog message:

```
*Jul 28 23:37:59.699: %GDOI-3-PSEUDO_TIME_TOO_OLD: Rekey received in group GET is too old
and fail PST check: my_pst is 22490 sec, peer_pst is 10026 sec, allowable_skew is 30 sec
```

ANN Message Pseudotime Handling in the Secondary Key Server

Cooperative key server announcement (ANN) messages are used to synchronize policy and group-member information between cooperative key servers.

The secondary key server handles ANN messages as follows:.

- The secondary key server calculates the allowable ANN message pseudotime as the lesser of the configured TBAR window size, that is, the value that was configured for it in the data plane, or 30 seconds.
- If the secondary key server receives an ANN message from the primary key server with a larger pseudotime, it does the following:
- It updates its pseudotime to the primary key server's value.
- If the pseudotime difference is larger than allowable, it generates the following syslog message:

```
*Jul 28 23:48:56.871: %GDOI-4-GDOI_ANN_TIMESTAMP_LARGE: COOP_KS ANN received from KS 10.0.8.1
 in group GET has pseudotime bigger than myself. Adjust to new pseudotime:
my_old_pst is 23147 sec, peer_pst is 30005 sec
```

- If the secondary key server receives an ANN message from the primary key server with a smaller pseudotime, it behaves as follows:
  - If the difference is within the allowable range, the secondary key server accepts it and updates its pseudotime to the primary key server's value.
  - If the difference exceeds the allowable range, it generates the following syslog message:

```
*Jul 28 23:42:12.603: %GDOI-4-GDOI_ANN_TIMESTAMP_TOO_OLD: COOP_KS ANN from KS 10.0.8.1 in
group GET is too old and fail PST check:
my_pst is 22743 sec, peer_pst is 103 sec, allowable_skew is 10 sec
```

If, after three retransmit requests, the secondary key server has still not received any ANN message with a valid pseudotime, it starts blocking new group-member registrations, as follows:

```
*Jul 28 23:38:57.859: %GDOI-5-COOP_KS_VALID_ANN_TIMER_EXPIRED: This sec-KS has NOT received
 an ANN with valid pseudotime for an extended period in group GET. It will block new group
 members registration temporarily until a valid ANN is received
*Jul 29 00:08:47.775: %GDOI-5-COOP_KS_BLOCK_NEW_GM_REGISTER: This key server temporarily
blocks group member with ip-addr 10.0.0.2 from registering in group GET as it has not
received an ANN with valid pseudotime for prolonged period
```

The secondary key server resumes its group member registration functionality if any of the following happens:

- It receives an ANN with a valid pseudotime from the primary key server.
- It becomes a primary key server itself.
- The **clear crypto gdoi group** command is executed on the secondary key server.

ANN Message Pseudotime Handling in the Primary Key Server

The primary key server handles ANN messages as follows:

- It calculates the allowable ANN message pseudotime as the lesser of the configured TBAR window size, that is, the value that was configured for it in the data plane, or 30 seconds.

- It accepts from the secondary key server ANN messages that have a smaller pseudotime but are within the allowable difference.
- It rejects ANN messages that have a smaller pseudotime but exceed the allowable difference.

During a network merge, the following conditions apply:

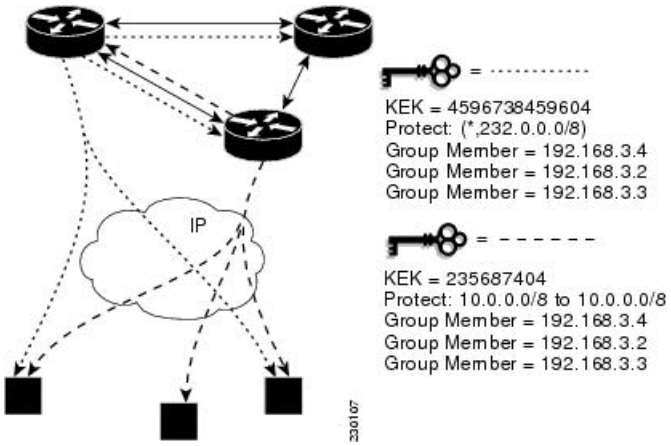
- The new primary key server always picks the larger pseudotime between the two key servers.
- If the difference is larger than the calculated allowable pseudotime difference, the new primary key server sends out rekeys to all group members to update their pseudotime. It also generates the following syslog messages:

```
*Jul 28 23:42:41.311: %GDOI-5-COOP_KS_ELECTION: KS entering election mode in group GET
(Previous Primary = NONE)
*Jul 28 23:42:41.311: %GDOI-4-GDOI_ANN_TIMESTAMP_LARGE: COOP_KS ANN received from KS 10.0.9.1
 in group GET has PST bigger than myself. Adjust to new pseudotime:
my_old_pst is 0 sec, peer_pst is 22772 sec
*Jul 28 23:43:16.335: %GDOI-5-COOP_KS_TRANS_TO_PRI: KS 10.0.8.1 in group GET transitioned
to Primary (Previous Primary = NONE)
*Jul 28 23:43:16.347: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group GET
from address 10.0.8.1 with seq # 1
```

Cooperative Key Server

The figure below illustrates cooperative key server key distribution. The text following the illustration explains the Cooperative Key Server feature.

Figure 11: Cooperative Key Server Key Distribution



Cooperative key servers provide redundancy to GET VPN. Multiple key servers are supported by GET VPN to ensure redundancy, high availability, and fast recovery if the primary key server fails. Cooperating GDOI key servers jointly manage the GDOI registrations for the group. Each key server is an active key server, handling GDOI registration requests from group members. Because the key servers are cooperating, each key server distributes the same state to the group members that register with it. Load balancing is achieved because each of the GDOI key servers can service a portion of the GDOI registrations.

The primary key server is responsible for creating and distributing group policy. When cooperative key server key distribution occurs, one key server declares itself as primary, creates a policy, and sends the policy to the other secondary key server. The secondary key server declares the primary key server as primary key server when it gets the policy and ends the election mode. The secondary key server now also blocks GM registration

while the cooperative key server key distribution is in progress. This change allows the cooperative key server distribution to become more efficient because it saves time. For example, the syslog warning message similar to the following is displayed during distribution:

```
00:00:16: %GDOI-5-COOP_KS_BLOCK_NEW_GM_REGISTER_ELECTION: This KS temporarily blocks GM
with ip-addr 10.0.4.1 from registering in group diffint as the KS election is underway
```

The primary key server periodically sends out (or broadcasts) group information updates to all other key servers to keep those servers in synchronization. If the secondary key servers somehow miss the updates, they contact the primary key server to directly request information updates. The secondary key servers mark the primary key server as unreachable (that is, “dead”) if the updates are not received for an extended period.

When a new policy is created on a primary key server, regardless of which key server a group member may be registered with, it is the responsibility of the primary key server to distribute rekey messages to GDOI group members.

In a cooperative-key-server setting, the rekey sequence number is synchronized between the primary and secondary key servers.

In a network merge, the key servers pick the larger of the rekey sequence numbers that they have between them.

If you are supporting more than 300 group members in your cooperative key server setup, you should increase the buffer size by using the **buffers huge size** command.

When an interface of the cooperative-key-server's address is shut, the network will split. If the interface is not the forwarding interface, which is the recommended configuration, rekeys will be sent to the group members from all of the key servers in the group. In this scenario, use the **no crypto gdoi group** *group name* command.

Announcement Messages

Announcement messages are secured by IKE Phase 1 and are sent as IKE notify messages. Authentication and confidentiality that are provided by IKE is used to secure the messaging between the key servers. Antireplay protection is provided by the sequence numbers in the announcement messages. Announcement messages are periodically sent from primary to secondary key servers.

Announcement messages include the components, described in the following sections that help maintain the current state.

Sender Priority of a Key Server

This value describes the priority of the sender, which is configurable using the CLI. The key server with the highest priority becomes the primary key server. If the priority values are the same, the key server with the highest IP address becomes the primary key server.

Maintaining the Role of the Sender

During the synchronization period, if the key servers are at geographically dispersed locations, they may suffer a network-partitioning event. If a network-partitioning event occurs, more than one key server can become the primary key server for a period of time. When the network is operating normally again and all the key servers find each other, they need to be told the current role of the sender so the key servers can attain their proper roles.

Request for a Return Packet Flag

All messages are defined as one-way messages. When needed, a key server can request the current state from a peer to find out its role or request the current state of the group.

Group Policies

The group policies are the policies that are maintained for a group, such as group member information and IPsec SAs and keys.

Antireplay functionalities and incorporated Cooperative announcement messages are supported. The primary key server updates the pseudotime value, sending it to all secondary key servers in the group. The secondary key servers should synchronize their SAR clocks to this updated value.

ANN Message Sequence Number Check Between Cooperative Key Servers

The following describes the sequence number check between cooperative key servers:

- Cooperative key servers drop any ANN message with a sequence number smaller than or equal to that of the last received ANN message.
- The ANN message is accepted if the sequence number is larger than that of the last received rekey message, no matter how large the difference.
- If a key server is reloaded, a new IKE session is created between the peers, and the reloaded key server's ANN sequence number will start with zero. In this case, the other side will accept the ANN message with any sequence number.

Change Key Server Role

In a network of cooperative key servers, the primary server is elected based on its highest priority at the time of election. The other key servers have secondary status. If the primary key server is detected as being dead or if its role changes, the **clear crypto gdoi ks coop role** command allows you to reset the cooperative role of the primary key server.

If the **clear crypto gdoi ks coop role** command is executed on a secondary key server, the election is triggered on that secondary key server although that server would most likely remain a secondary key server because there has been an elected primary key server. However, if the **clear crypto gdoi ks coop role** command is executed on the primary key server, the primary key server is reassigned to a secondary role, and as a result, a new election that involves all the key servers is triggered. If the previous primary server has the highest priority (of all the key servers), it again becomes the primary server. If the previous primary server does not have the highest priority, the server having the highest priority is elected as the new primary server.

Receive Only SA

For multicast traffic using the GDOI protocol, bidirectional SAs are installed. The Receive Only feature enables an incremental deployment so that only a few sites can be verified before bringing up an entire network. To test the sites, one of the group members should send encrypted traffic to all the other group members and have them decrypt the traffic and forward the traffic “in the clear.” Receive Only SA mode allows encryption in only the inbound direction for a period of time. (See the steps for the Receive Only SA process.) If you configure the **sa receive-only** command on the key server, Steps 2 and 3 happen automatically.

1. Mark IPsec SAs as “receive-only” on the GDOI key server.

This action allows the group members to install SAs in the inbound direction only. Receive-only SAs can be configured under a crypto group. (See the "Configuring the Group ID Server Type and SA Type" section.)

1. Mark GDOI TEK payloads as "receive only."

If the **sa receive-only** command is configured, all TEKs under this group are going to be marked "receive only" by the key server when they are sent to the group member.

1. Install one-way IPsec flows.

Every time a GDOI group member receives an IPsec SA from the key server that is marked as "receive only," the group member installs this IPsec SA only in the inbound direction rather than in both incoming and outgoing directions.

1. Test individual group members using the following local-conversion commands:
2. **crypto gdoi gm ipsec direction inbound optional**
3. **crypto gdoi gm ipsec direction both**

First, individually convert each of the group members to passive mode (this change tells the outbound check that there is a valid SA) and then to bidirectional mode.

1. Globally convert from "receive only" to "receive and send."

The following method can be used when the testing phase is over and "receive only" SAs have to be converted to bidirectional SAs.

Global Conversion

Remove the **sa receive-only** command under the group. Removing the **sa receive-only** command creates new IPsec SAs for this group and causes a rekey. On receipt, group members reinstall the SA in both directions and begin to use it in passive mode. Because the SA cannot remain in passive mode forever, the group members change those SAs to receive or send mode if there is no rekey in 5 minutes. The conversion from passive mode to bidirectional encryption mode is automatic and does not require the administrator to do anything.

Passive SA

The Passive SA feature allows you to configure a group member so that it is in passive mode permanently. By using the Passive SA feature, you will avoid having to use the **crypto gdoi gm ipsec direction inbound optional** privileged EXEC command, which is not persistent after a router reload and can be overridden by key server configuration from a rekey. Having the group member in passive mode benefits network testing and debugging during migration to GET VPN, and it provides complete encryption protection during the migration. The group-member passive-mode configuration has higher priority over a key server configuration. The **crypto gdoi gm ipsec direction inbound optional** privileged EXEC command can override the configuration until the next rekey, which will bring back the group member and key server configuration.

To configure the Passive SA feature, see the "Configuring Passive SA" section.

Enhanced Solutions Manageability

Several **show** and **debug** commands are supported to help verify functionality. See the "Activating Fail-Close Mode" section for details.

Support with VRF-Lite Interfaces

The VRF-Lite application supports segmentation of traffic in the control and forwarding planes by keeping the routing tables separate for each user group (or VPN) and forwarding the traffic on the associated or dedicated interfaces of each user group.

There are some deployment scenarios in which remote sites that are connecting to an MPLS VPN network might be extending segmentation from a campus to the WAN. In such an extended segmentation case, a CE-PE interface on a CE (group member or key server) device "bounds" to its associated virtual routing and forwarding (VRF) instance. This VRF interface connects to an MPLS PE device where it is directly mapped to its associated Border Gateway Protocol (BGP) VRF process, in which case the crypto map is applied to a VRF interface. No other configuration changes are necessary.

Authentication Policy for GM Registration

GMs can authenticate to the key server at registration time using preshared keys or Public Key Infrastructure (PKI). Preshared keys are easy to deploy but must be managed proactively. We recommend that you deploy a peer-based preshared key instead of defining a default key (the key defined with an address of 0.0.0.0) for all the devices in the network. Preshared keys should be updated regularly (every few months).

Note A preshared key can be updated on a key server-group member (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 re-register to each ordered set of key servers using the newly assigned preshared key.

PKI uses its infrastructure to overcome the key management difficulties encountered when preshared keys are used. 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, key server capacity, design, and placement become important.

For added security, GET VPN also supports GM authorization using either preshared keys or PKI. For more information, see the "GET VPN Authorization" section.

GET VPN GM Authorization

GET VPN GM authorization can be done using preshared keys or PKI. It is a best practice to turn on GET VPN authorization. When a key server serves multiple GDOI groups, key server authorization is required to prevent a GM in one group from requesting keys and policies from another group. The ISAKMP authorization confirms that the GM is allowed to request GDOI attributes from the key server while the GDOI authorization confirms that the GM is allowed to request GDOI attributes from a specific group configured in the key server.

GDOI authorization is based on the ISAKMP identity sent by a GM. If a GM sends an IP address as an identity, then only an authorization address is used for authorization. If a GM sends a distinguished name (DN) or hostname, then an authorization identity is used. 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), you must specify **deny any** in the authorization address.

GM Authorization Using Preshared Keys

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

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
```

GM Authorization Using PKI

GET VPN supports GM authorization using the commonly used DN or fully qualified domain name (FQDN) when PKI is used. The **authorization identity** command is used to activate GM authorization. A crypto identity matching certain fields in the GM certificate (typically—organizational unit [OU]) can be defined and applied to the GET VPN group configuration. Use the **crypto identity** command to define a crypto identity.

Any GM whose certificate credentials match the ISAKMP identity is authorized and can register to the key server. For example, if all GM certificates are issued with OU=GETVPN, a key server 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 key server.

If authorization is unsuccessful, the following syslog message is generated:

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

Rekey Functionality in Protocol Independent Multicast-Sparse Mode

Multicast rekeying can be used with all modes of multicast. The **rekey retransmit** command should be used whenever the Protocol Independent Multicast-sparse mode (PIM-SM) is configured because the PIM-SM shortest path tree (SPT) can be torn down if it does not receive continuing traffic. When traffic resumes, PIM-SM must reestablish the SPT. Retransmitting rekey packets increases the chance that group members receive the rekeys when PIM-SM is setting up the SPT.

Fail-Close Mode

Until a group member registers with a key server, traffic passing through the group member is not encrypted. This state is called “fail open.” To prevent unencrypted traffic from passing through a group member before that member is registered, you can configure the Fail-Close feature. If the feature is configured, an implicit “permit ip any any” policy is installed, and all unencrypted traffic passing through the group member is dropped (this state is called fail-close mode).

The fail-close function can also be achieved by configuring an interface ACL. However, the Fail-Close feature is more manageable and is easier to implement than ACL lists.

If you configure the Fail-Close feature, you can still allow specific unencrypted traffic to pass through the group member by configuring the **match address** command (**match address** {*access-list-number* | *access-list-name*}). This explicit “deny” ACL is added before the implicit “permit ip any any” to allow denied (unencrypted) traffic to pass through the group member.

After the group member has successfully completed its registration, the fail-close policy, both explicit and implicit, is removed, and the group member behaves as it did before the Fail-Close feature was configured.

Guidelines for Using the Fail-Close Feature

When you are configuring a crypto map to work in fail-close mode, you must be careful. If the fail-close ACL is defined improperly, you may lock yourself out of the router. For example, if you use Secure Shell (SSH) to log in to the router through the interface with the crypto map applied, you have to include the **deny tcp**

any eq *port* **host** *address* command line under the fail-close ACL. You may also need to include the routing protocol that the router is using (such as **deny ospf any any**) to find the path to the key server. It is suggested that you configure fail-close and its ACL first, and then verify the fail-close ACL using the **show crypto map gdoi fail-close** *map-name* command. After you have checked your fail-close ACL and are confident that it is correct, you can make the crypto map work in the fail-close mode by configuring the **activate** command. Fail-close is not activated until you have configured the **activate** command.

The fail-close ACL is configured from the group-member perspective. The fail-close ACL is configured on group member as follows:

access-list 125 deny ip host *host1-ip-addr host2-ip-addr*

In fail-close mode, all IP traffic from host1 to host2 will be sent out by group member 1 in clear text. In addition, the inbound mirrored traffic (that is, IP traffic from host2 to host1) is also accepted by GM1 in clear text.

Note All IP traffic matching deny entries are sent out by the group member in clear text.

The inbound traffic is matched to the mirrored access list.

The fail-close access list follows the same rules as the group member access list. For more information, see the "Group Member Access Control List" section.

You need not configure the **deny udp any eq 848 any eq 848** command to make the GDOI registration go through. The code itself checks whether it is a GDOI packet for a particular group member from the key server to which it is configured. If it is a GDOI packet for this group member, the packet is processed. But for a scenario in which the key server is behind group member 1, if group member 1 cannot register successfully with the key server, other group members also will not be able to register unless an explicit **deny udp any eq 848 any eq 848** command line is configured for group member 1. However, if the Fail-Close feature is properly configured, even if a group member fails to register with a key server, you will be able to ensure that no unwanted traffic can go out “in the clear.” But you can allow specified traffic to go out in the clear, in which case registration packets from other group members will be able to reach the key server through group member 1 even if it fails to get registered.

For information on configuring fail-close mode, see the "Activating Fail-Close Mode" section.

To verify whether fail-close mode is activated, use the **show crypto map gdoi fail-close** command.

Create MIB Object to Track a Successful GDOI Registration

The routing plane and crypto plane for GET VPN must be synchronized to avoid routing black holes. A GET VPN routing black hole occurs during the following situations:

- GMs fail to register to a KS that has no active TEKs to encrypt or decrypt traffic.
- GMs TEK SAs have expired but do not receive new keys from KS through rekey or reregistration.
- GMs receive rekeys from KS, but errors occur when installing SAs to a crypto engine.

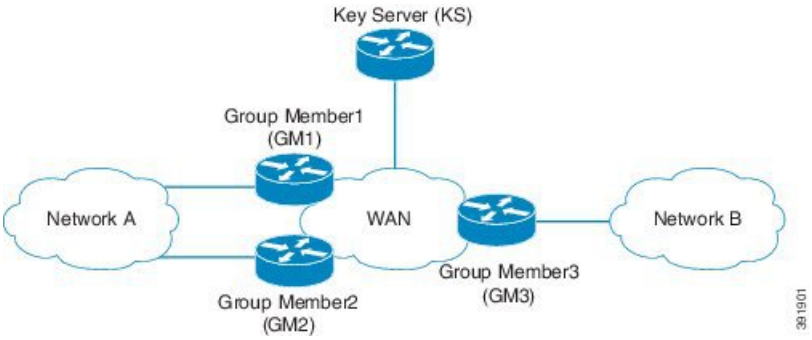
The Create MIB Object to Track a Successful GDOI Registration feature introduces a new MIB object in the GDOI MIB to indicate the number of active TEKs in a group.

GET VPN Routing Awareness for BGP

The routing plane and crypto plane for GET VPN must synchronize to avoid routing black holes. When a group member (GM) successfully registers to a key server (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 following diagram explains the creation of a black hole.

Figure 12: Routing Black Hole Creation



1. Group Member1, Group Member2, Group Member3 boot up and establish a routing adjacency with the WAN.
2. Group Member1 and Group Member2 advertises the prefix for Network A into the WAN. The preferred path for traffic from Network B to Network A is through Group Member1.
3. Group Member3 advertises Network B into the WAN. The preferred path for traffic Network A to Network B is through Group Member1
4. KS defines the security to protect all traffic between Network A and Network B
5. Group Member1 and Group Member3 (as well as Group Member2) successfully obtain security keys from KS and protect all traffic between Network A <-> Network B.
6. Group Member1 fails to receive updated keys or policy and fails to reregister to a KS while Group Member2 and Group Member3 successfully obtain keys.
7. Routing protocols continue to prefer the path through Group Member1 for all Network A between Network B traffic.
8. Group Member1 drops all traffic flowing between Network A and Network B because the policy/keys are invalid.

When the host in Network B sends traffic to a host in Network A, the traffic will be encrypted by Group Member3 and sent to Network A via Group Member1 (the preferred path). However, Group Member1 will drop the packet because it has no policy or current keys to decrypt traffic. As a result, the traffic is dropped and a black hole is created. Likewise, when a host in Network A sends traffic to a host in Network B, the traffic will be directed to Group Member1 (the preferred path) and dropped due to lack of policy or current keys in Group Member1. The appropriate behavior is for the traffic to be diverted and rerouted via Group Member2 while Group Member1 has no policy or keys.

The GET VPN Routing Awareness for BGP 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.

Bidirectional Conditional Route Filtering

The bidirectional conditional route filtering supports different routing protocols, such as BGP, OSPF, EIGRP, RIPv2, etc. The EOT tracks the GET VPN GM crypto status and conditionally enables or disables specific route-map entries based on the EOT value. The following is a sample configuration to monitor the GET VPN GM crypto state.

```
route-map bgp-policy-out permit 10
 match ip address register-int-Only
route-map bgp-policy-out permit 20
 match track 99
 match ip address orig_route_map_acl_out
route-map bgp-policy-out deny 30

route-map bgp-policy-in permit 10
 match ip address noc
route-map bgp-policy-in permit 20
 match track 99
 match ip address orig_route_map_acl_in
route-map bgp-policy-in deny 30

ip access-list standard noc
 permit 1.1.1.0                                  <---- NOC subnet with Keyserver (KS)
ip access-list standard register-int-Only
 permit 2.2.2.2                                  <---- registration interface ip of the
GM itself
ip access-list standard orig_route_map_acl_in       <---- original inbound route-map ACL
 permit a.b.c.d
 permit .......
ip access-list standard orig_route_map_acl_out      <---- original outbound route-map ACL
 permit e.f.g.h
 permit .......

router bgp 64600
 no synchronization
 bgp router-id xxxxxxxx
 bgp log-neighbor-changes
 network xxxxxxxxxx mask 255.255.255.255
 network xxxxxxxxxx mask 255.255.255.252
 neighbor xxxxxxxxxx remote-as 65000
 neighbor xxxxxxxxxx description PE
 neighbor xxxxxxxxxx route-map bgp-policy-in in
 neighbor xxxxxxxxxx route-map bgp-policy-out out
```

In the above example, the **match track 99** command is specified to monitor the GET VPN GM crypto state. If GM works properly, the **match track 99** command returns a value *true* and the GM advertises or receives the following routes:

- Outbound—The routes to reach the GM registration interface and the routes permitted by inbound route map access control list (ACL) “orig_route_map_acl_out.”
- Inbound—The routes to reach the NOC and the routes permitted by outbound route map ACL “orig_route_map_acl_in” received from peers.

On the other hand, if GM does not work properly, the **match track 99** command returns a value *false* and the GM advertises or receives the following routes only:

- Outbound—The routes to reach the GM registration interface.
- Inbound—The routes to reach the NOC subnet.

Cisco Group Encrypted Transport VPN System Logging Messages

The table below lists GET VPN system logging (also called syslog) messages and explanations.

Table 3: GET VPN System Logging 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 unauthorized 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.

Message	Explanation
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_DELETE	A GM has been deleted in a group from a KS.
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_NO_IPSEC_FLOWS	The hardware limitation for IPsec flow limit reached. Cannot create any more IPsec SAs.

Message	Explanation
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_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.

Message	Explanation
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.

Message	Explanation
UNREGISTERED_INTERFACE	Receiving registration from unregistered interface. Stop processing it.
UNSUPPORTED_TEK_PROTO	Unexpected TEK protocol.

How to Configure Cisco Group Encrypted Transport VPN

Configuring a Key Server

Prerequisites

Before creating the GDOI group, you must first configure IKE and the IPsec transform set, and you must create an IPsec profile. For information about how to configure IKE and the IPsec transform set and to create an IPsec profile, see the "Related Documents" subsection of the "Additional References" section.

Configuring RSA Keys to Sign Rekey Messages

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

To configure RSA keys that will be used to sign rekey messages, perform the following steps. Omit this subtask if rekey is not in use.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto key generate rsa general-keys label** *name-of-key*

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**crypto key generate rsa general-keys label** *name-of-key* **Example:** `Router(config)# crypto key generate rsa general-keys label mykeys`	Generates RSA keys that will be used to sign rekey messages. You are prompted to confirm the length (in bits) of the keys to be generated. Length of less than 2048 is not recommended.

What to Do Next

Configure the group ID, server type, and SA type. (See the "Configuring the Group ID Server Type and SA Type" section.)

Configuring the Group ID Server Type and SA Type

For a large number of sites, it is better to take precautions and add functionality incrementally, especially when migrating from any other encryption solutions like Dual Multipoint VPN (DMVPN). For example, instead of setting up all the CPE devices to encrypt the traffic bidirectionally, it is possible to configure one-way encryption so that only one or fewer members of a group are allowed to send encrypted traffic. Others are allowed to receive only encrypted traffic. After the one-way encryption is validated for one or a few members, bidirectional encryption can be turned on for all the members. This "inbound only" traffic can be controlled using the **sa receive only** command under a crypto group.

To configure the group ID, server type, and SA type, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server local**
6. **sa receive-only**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupname`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Router(config-gdoi-group)# identity number 3333`
	Example:
	`Router(config-gdoi-group)# identity address ipv4 209.165.200.225`
		Identifies a GDOI group number or address.

Step 5	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**sa receive-only** **Example:** `Router(config-local-server)# sa receive-only`	Specifies that an IPsec SA is to be installed by a group member as "inbound only."

What to Do Next

Remove the receive-only configuration on the key server so that the group members are now operating in bidirectional receive and send mode.

Configuring the Rekey

This section includes the following optional tasks:

Rekey is used in the control plane by the key server to periodically refresh the policy and IPsec SAs of the group. On the group-member side, instead of fully re-registering when timers expire for any other reasons, refreshing the registration with a rekey is more efficient. The initial registration is always a unicast registration.

The key server can be configured to send rekeys in unicast or multicast mode. The rekey transport mode is determined by whether the key server can use IP multicast to distribute the rekeys. If multicast capability is not present within the network of the customer, the key server will have to be configured to send rekeys using unicast messages.

Additional options for rekey use the **rekey authentication**, **rekey retransmit**, and **rekey address ipv4** commands. If unicast transport mode is configured, the **source address** command will have to be included to specify the source address of this unicast rekey message.

Multicast is the default transport type for rekey messages. The following bulleted items explain when to use rekey transport type multicast or unicast:

- If all members in a group are multicast capable, do not configure the **rekey transport unicast** command. The **no rekey transport unicast** command is not needed if the rekey transport type "unicast" was not configured previously under this group because multicast rekeys are on by default.
- If all members in a group are unicast, use the **rekey transport unicast** command.
- If you have mixed members in a group (that is, the majority are multicast, but a few are unicast), do not configure the **rekey transport unicast** command. The rekeys will be distributed using multicast to the majority of group members. The remainder of the group members that do not receive the multicast messages (unicast group members) will have to re-register to the key server when their policies expire. Mixed mode (that is, unicast and multicast rekey mode) is not supported.

If the **no rekey transport unicast** command is used, members in the GDOI group that are unable to receive the multicast rekey messages need to re-register with the key server to get the latest group policies. The re-registration forces the default transport type to multicast. If no transport type was configured previously, the multicast transport type will apply by default.

Prerequisites

Before configuring the **rekey authentication** command, you must have configured the router to have an RSA key generated using the **crypto key generate rsa** command and **general-keys** and **label** keywords (for example, "crypto key generate rsa general-key label my keys").

Configuring a Unicast Rekey

In the configuration task table, the address "ipv4 10.0.5.2" specifies the interface on the key server by which the unicast or multicast rekey messages are sent. This address is required for unicast rekeys, but it is optional for multicast rekeys. For multicast rekeys, the source address of the key server can be retrieved from the rekey ACL.

To configure a unicast rekey, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server local**
6. **rekey transport unicast**
7. **rekey lifetime seconds** *number-of-seconds*
8. **rekey retransmit** *number-of-seconds* **number** *number-of-retransmissions*
9. **rekey authentication mypubkey rsa** *key-name*
10. **address ipv4** *ipv4-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Router> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupname`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Router(config-gdoi-group)# identity number 3333`
	Example:
	`Router(config-gdoi-group)# identity address ipv4 209.165.200.225`
		Identifies a GDOI group number or address.

Step 5	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**rekey transport unicast** **Example:** `Router(config-local-server)# rekey transport unicast`	Configures unicast delivery of rekey messages to group members.
Step 7	**rekey lifetime seconds** *number-of-seconds* **Example:** `Router(gdoi-local-server)# rekey lifetime seconds 300`	(Optional) Limits the number of seconds that any one encryption key should be used. • If this command is not configured, the default value of 86,400 seconds takes effect.
Step 8	**rekey retransmit** *number-of-seconds* **number** *number-of-retransmissions* **Example:** `Router(gdoi-local-server)# rekey retransmit 10 number 3`	(Optional) Specifies the number of times the rekey message is retransmitted. • If this command is not configured, there will be no retransmits.

	Command or Action	Purpose
Step 9	**rekey authentication mypubkey rsa** *key-name* **Example:** `Router(gdoi-local-server)# rekey authentication mypubkey rsa mykeys`	(Optional) Specifies the keys to be used for a rekey to GDOI group members. • This command is optional if rekeys are not required. If rekeys are required, this command is required.
Step 10	**address ipv4** *ipv4-address* **Example:** `Router(gdoi-local-server)# address ipv4 209.165.200.225`	(Optional) Specifies the source information of the unicast rekey message. • If rekeys are not required, this command is optional. If rekeys are required, this command is required.

Configuring a Multicast Rekey

To configure a multicast rekey, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server local**
6. **rekey address ipv4** {*access-list-name* | *access-list-number*}
7. **rekey lifetime seconds** *number-of-seconds*
8. **rekey retransmit** *number-of-seconds* **number** *number-of-retransmissions*
9. **rekey authentication** {**mypubkey** | **pubkey**} **rsa** *key-name*
10. **exit**
11. **exit**
12. **access-list** *access-list-number* {**deny** | **permit**} **udp host** *source* [*operator*[*port*]] **host** *source* [*operator*[*port*]]
13. **interface** *type slot/ port*
14. **ip igmp join-group** *group-address* [**source** *source-address*]

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupname`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Router(config-gdoi-group)# identity number 3333`
	Example:
	`Router(config-gdoi-group)# identity address ipv4 209.165.200.225`
		Identifies a GDOI group number or address.

Step 5	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**rekey address ipv4** {*access-list-name* | *access-list-number*} **Example:** `Router(gdoi-local-server)# rekey address ipv4 121`	Defines to which multicast subaddress range group members will register.
Step 7	**rekey lifetime seconds** *number-of-seconds* **Example:** `Router(gdoi-local-server)# rekey lifetime seconds 300`	(Optional) Limits the number of seconds that any one encryption key should be used. • If this command is not configured, the default value of 86,400 seconds takes effect.
Step 8	**rekey retransmit** *number-of-seconds* **number** *number-of-retransmissions* **Example:** `Router(gdoi-local-server)# rekey retransmit 10 number 3`	(Optional) Specifies the number of times the rekey message is retransmitted. • If this command is not configured, there will be no retransmits.
Step 9	**rekey authentication** {**mypubkey** | **pubkey**} **rsa** *key-name*	(Optional) Specifies the keys to be used for a rekey to GDOI group members.

	Command or Action	Purpose
	Example: `Router(gdoi-local-server)# rekey authentication mypubkey rsa mykeys`	• This command is optional if rekeys are not required. If rekeys are required, this command is required.
Step 10	**exit** **Example:** `Router(gdoi-local-server)# exit`	Exits GDOI server local configuration mode.
Step 11	**exit** **Example:** `Router(config-gdoi-group)# exit`	Exits GDOI group configuration mode.
Step 12	**access-list** *access-list-number* {**deny** | **permit**} **udp host** *source* [*operator*[*port*]] **host** *source* [*operator*[*port*]] **Example:** `Router(config)# access-list 121 permit udp host 10.0.5.2 eq 848 host 239.0.1.2 eq 848`	Defines an extended IP access list.
Step 13	**interface** *type slot/ port* **Example:** `Router(config)# interface gigabitethernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 14	**ip igmp join-group** *group-address* [**source** *source-address*] **Example:** `Router(config-if)# ip igmp join-group 232.2.2.2 source 10.1.1.1`	Configures an interface on the router to join the specified group or channel. **Note** Use this command to manually join the stream when the key server is not reachable via the same interface as the one configured with the crypto map.

Configuring Group Member ACLs

All IP traffic matching deny entries are sent out by the group member in clear text. The inbound traffic is matched to the mirrored access list.

Note The recommended method to add or delete an entry in the Group Member ACL is to first create a copy of the existing Group Member ACL with a different name and then add or delete the entry in this new ACL, after which, you should replace the existing group member ACL under the GDOI crypto map with the newly created Group Member ACL.

To configure group member ACLs, perform this task (note that a group member access list can contain only deny statements).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **deny ip host** *source* **host** *source*
4. **access-list** *access-list-number* **permit ip** *source*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* **deny ip host** *source* **host** *source* **Example:** `Router(config)# access-list 101 deny ip host 10.0.0.1 host 10.0.0.2`	Defines a denied IP access list.
Step 4	**access-list** *access-list-number* **permit ip** *source* **Example:** `Router(config)# access-list 103 permit ip 209.165.200.225 0.255.255.255 10.20.0.0 0.255.255.255`	Defines an allowed IP access list.

What to Do Next

The access list defined in Step 4 is the same one that should be used to configure the SA. See the "Configuring the IPsec SA" section.

Configuring an IPsec Lifetime Timer

To configure an IPsec lifetime timer for a profile, perform the following steps. If this configuration task is not performed, the default is the maximum IPsec SA lifetime of 3600 seconds. The TEK lifetime value should be more than 900 seconds.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *name*

4. **set security-association lifetime seconds** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *name* **Example:** `Router(config)# crypto ipsec profile profile1`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec routers and enters crypto ipsec profile configuration mode.
Step 4	**set security-association lifetime seconds** *seconds* **Example:** `Router(ipsec-profile)# set security-association lifetime seconds 2700`	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPsec SAs.

What to Do Next

Configure the IPsec SA. See the “Configuring IPsec SA” section.

Configuring an ISAKMP Lifetime Timer

To configure an ISAKMP lifetime timer, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp policy** *priority*
4. **lifetime** *seconds*

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 1`	Defines an IKE policy and enters ISAKMP policy configuration mode.
Step 4	**lifetime** *seconds* **Example:** `Router(config-isakmp-policy)# lifetime 86400`	Specifies the lifetime of an IKE SA.

Configuring the IPsec SA

If time-based antireplay is configured on the key server but the group member is not capable of supporting it, the GDOI-3-GM_NO_CRYPTO_ENGINE syslog message is logged to the group member. See the "Cisco Group Encrypted Transport VPN System Logging Messages" section for a list of system error messages.

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

To configure the IPsec SA, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name* *transform* [*transform2...transform4*]
4. **crypto ipsec profile** *ipsec-profile-name*
5. **set transform-set** *transform-set-name*
6. **exit**
7. **crypto gdoi group** *group-name*
8. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
9. **server local**
10. **sa ipsec** *sequence-number*
11. **profile** *ipsec-profile-name*
12. **match address ipv4** {*access-list-number* | *access-list-name*}
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform* [*transform2...transform4*] **Example:** `Router(config)# crypto ipsec transform-set gdoi-trans esp-aes esp-sha-hmac`	Defines a transform set--an acceptable combination of security protocols and algorithms.
Step 4	**crypto ipsec profile** *ipsec-profile-name* **Example:** `Router(config)# crypto ipsec profile profile1`	Defines an IPsec profile and enters crypto ipsec profile configuration mode.
Step 5	**set transform-set** *transform-set-name* **Example:** `Router(ipsec-profile)# set transform-set transformset1`	Specifies which transform sets can be used with the crypto map entry.
Step 6	**exit** **Example:** `Router(ipsec-profile)# exit`	Exits IPsec profile configuration mode.
Step 7	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupname`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 8	Enter one of the following commands: • **identity number** *number* • **identity address ipv4** *address* **Example:** `Router(config-gdoi-group)# identity number 3333` **Example:**	Identifies a GDOI group number or address.

	Command or Action	Purpose
	`Router(config-gdoi-group)# identity address ipv4 209.165.200.225`	
Step 9	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 10	**sa ipsec** *sequence-number* **Example:** `Router(gdoi-local-server)# sa ipsec 1`	Specifies the IPsec SA policy information to be used for a GDOI group and enters GDOI SA IPsec configuration mode.
Step 11	**profile** *ipsec-profile-name* **Example:** `Router(gdoi-sa-ipsec)# profile gdoi-p`	Defines the IPsec SA policy for a GDOI group.
Step 12	**match address ipv4** {*access-list-number* | *access-list-name*} **Example:** `Router(gdoi-sa-ipsec)# match address ipv4 102`	Specifies an IP extended access list for a GDOI registration.
Step 13	**end** **Example:** `Router(gdoi-sa-ipsec)# end`	Exits GDOI SA IPsec configuration mode and returns to privileged EXEC mode.

What to Do Next

Replay should be configured. If replay is not configured, the default is counter mode.

Configuring Time-Based Antireplay for a GDOI Group

To configure time-based antireplay for a GDOI group, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **identity number** *policy-name*
5. **server local**
6. **address** *ip-address*
7. **sa ipsec** *sequence-number*
8. **profile** *ipsec-profile-name*

9. **match address** {**ipv4** *access-list-number* | *access-list-name*}
10. **replay counter window-size** *seconds*
11. **replay time window-size** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroup1`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**identity number** *policy-name* **Example:** `Router(config-gdoi-group)# identity number 1234`	Identifies a GDOI group number.
Step 5	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**address** *ip-address* **Example:** `Router(config-server-local)# address 209.165.200.225`	Sets the source address, which is used as the source for packets originated by the local key server.
Step 7	**sa ipsec** *sequence-number* **Example:** `Router(config-server-local)# sa ipsec 1`	Specifies the IPsec SA and enters GDOI SA IPsec configuration mode.
Step 8	**profile** *ipsec-profile-name* **Example:** `Router(gdoi-sa-ipsec)# profile test1`	Defines the IPsec SA policy for a GDOI group.

	Command or Action	Purpose
Step 9	**match address** {**ipv4** *access-list-number* | *access-list-name*} **Example:** `Router(gdoi-sa-ipsec)# match address ipv4 101`	Specifies an IP extended access list for a GDOI registration.
Step 10	**replay counter window-size** *seconds* **Example:** `Router(gdoi-sa-ipsec)# replay counter window-size 512`	Turns on counter-based antireplay protection for traffic defined inside an access list using GDOI if there are only two group members in a group. **Note** The behavior caused by this command and that caused by the **replay time window-size** command are mutually exclusive. You can configure either one without configuring the other.
Step 11	**replay time window-size** *seconds* **Example:** `Router(gdoi-sa-ipsec)# replay time window-size 1`	Sets the window size for antireplay protection using GDOI if there are more than two group members in a group. **Note** The behavior caused by this command and that caused by the **replay counter window-size** command are mutually exclusive. You can configure either one without configuring the other.

Configuring Passive SA

To configure passive SA (to put the group member in passive mode), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **identity** *name*
5. **passive**
6. **server address ipv4** {*address* | *hostname*}

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group group1`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**identity** *name* **Example:** `Router(config-gdoi-group)# identity 2345`	Sets the identity to the crypto map.
Step 5	**passive** **Example:** `Router(config-gdoi-group)# passive`	Puts the group member into passive mode.
Step 6	**server address ipv4** {*address* | *hostname*} **Example:** `Router(config-gdoi-group)# server address ipv4 209.165.200.225`	Specifies the address of the server that a GDOI group is trying to reach.

Resetting the Role of the Key Server

To reset the cooperative role of the primary key server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **clear crypto gdoi ks coop role**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto gdoi ks coop role** **Example:** `Router# clear crypto gdoi ks coop role`	Resets the cooperative role of the key server.

Configuring a Group Member

To configure a group member, perform the following subtasks:

Configuring the Group Name ID Key Server IP Address and Group Member Registration

To configure the group name, ID, key server IP address, and group member registration, perform the following steps. You can configure up to eight key server addresses.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. Do one of the following:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server address ipv4** *address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Do one of the following:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Router(config-gdoi-group)# identity number 3333`
	Example:
	`Router(config-gdoi-group)# identity address ipv4 209.165.200.225`
		Identifies a GDOI group number or address.

	Command or Action	Purpose
Step 5	**server address ipv4** *address* **Example:** `Router(config-gdoi-group)# server address ipv4 209.165.200.225`	Specifies the address of the server a GDOI group is trying to reach. • To disable the address, use the **no** form of the command.

What to Do Next

Configure a crypto map. See the "Creating a Crypto Map Entry" section.

Creating a Crypto Map Entry

To create a crypto map entry and associate a GDOI group to it, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto map** *map-name seq-num* **gdoi**
4. **set group** *group-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name seq-num* **gdoi** **Example:** `Router(config)# crypto map mymap 10 gdoi`	Enters crypto map configuration mode and creates or modifies a crypto map entry.
Step 4	**set group** *group-name* **Example:** `Router(config-crypto-map)# set group group1`	Associates the GDOI group to the crypto map.

What to Do Next

Apply the crypto map to an interface to which the traffic has to be encrypted. See the "Applying the Crypto Map to an Interface to Which the Traffic Must Be Encrypted" section.

Applying the Crypto Map to an Interface to Which the Traffic Must Be Encrypted

To apply the crypto map to an interface to which the traffic must be encrypted, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot* / *port*
4. **crypto map** *map-name* **redundancy** *standby-group-name* **stateful**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type slot* / *port* **Example:** `Router(config)# interface gigabitethernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**crypto map** *map-name* **redundancy** *standby-group-name* **stateful** **Example:** `Router(config-if)# crypto map map1`	Applies the crypto map to the interface.

Activating Fail-Close Mode

Fail-close mode prevents unencrypted traffic from passing through a group member before that member is registered with a key server.

To configure a crypto map to work in fail-close mode, perform the following steps:

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **crypto map** *map-name* **gdoi fail-close**
4. **match address** {*access-list-number* | *access-list-name*}
5. **activate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto map** *map-name* **gdoi fail-close** **Example:** `Router(config)# crypto map map1 gdoi fail-close`	Specifies that the crypto map is to work in fail-close mode and enters crypto map fail-close configuration mode.
Step 4	**match address** {*access-list-number* | *access-list-name*} **Example:** `Router(crypto-map-fail-close)# match address 133`	(Optional) Specifies an ACL for a GDOI registration.
Step 5	**activate** **Example:** `Router(crypto-map-fail-close)# activate`	Activates fail-close mode.

Configuring Acceptable Ciphers or Hash Algorithms for KEK

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

To configure the ciphers and hash algorithms for KEK to be allowed by the GM, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*

4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server address ipv4** *address*
6. **client rekey encryption** *cipher* [... [*cipher*]]
7. **client rekey hash** *hash*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Router(config-gdoi-group)# identity number 3333`
	Example:
	`Router(config-gdoi-group)# identity address ipv4 10.2.2.2`
		Identifies a GDOI group number or address.

Step 5	**server address ipv4** *address* **Example:** `Router(config-gdoi-group)# server address ipv4 10.0.5.2`	Specifies the address of the server a GDOI group is trying to reach. • To disable the address, use the **no** form of the command.
Step 6	**client rekey encryption** *cipher* [... [*cipher*]] **Example:**	Sets the client acceptable rekey ciphers for the KEK.

	Command or Action	Purpose
	`Router(config-gdoi-group)# client rekey encryption aes 128 aes 192 aes 256`	
Step 7	**client rekey hash** *hash* **Example:** `Router(config-gdoi-group)# client rekey hash sha`	Sets the client acceptable hash algorithm for KEK.
Step 8	**end** **Example:** `Router(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Configuring Acceptable Transform Sets for TEK

To configure the transform sets used by TEKs for data encryption or authentication to be allowed by the GM, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name transform* [*transform2...transform4*]
4. **exit**
5. **crypto gdoi group** *group-name*
6. **client transform-sets** *transform-set-name1* [... [*transform-set-name6*]]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name transform* [*transform2...transform4*] **Example:**	Defines a transform set—an acceptable combination of security protocols and algorithms—and enters crypto transform configuration mode.

	Command or Action	Purpose
	`Router(config)# crypto ipsec transform-set g1 esp-aes 192 esp-sha-hmac`	
Step 4	**exit** **Example:** `Router(cfg-crypto-trans)# exit`	Exits crypto transform configuration mode.
Step 5	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 6	**client transform-sets** *transform-set-name1* [... [*transform-set-name6*]] **Example:** `Router(config-gdoi-group)# client transform-sets g1`	Specifies the acceptable transform-set tags used by TEK for data encryption and authentication. • You can specify up to six transform-set tags.
Step 7	**end** **Example:** `Router(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Tracking the Group Member Crypto State

Perform this task to track the crypto state of the group member (GM) using the configured Enhanced Object Tracker (EOT) stub-object ID.

Before you begin

You must configure an Enhanced Object Tracking (EOT) by creating a stub-object and assign the object with a tracking ID to monitor the GDOI MIB. The following is a sample configuration in which, tracking ID 99 is assigned to the stub-object.

```
event manager applet test1
 event snmp oid <new GDOI MIB object> .....
 action 2.0 track set 99 state up

track 99 stub-object
 delay up 60
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **client status active-sa track** *tracking-number*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**client status active-sa track** *tracking-number* **Example:** `Device(config-gdoi-group)# client status active-sa track 99`	Enables the tracking for the stub-object. In this example, a GM will set the stub-object 99 to state "UP" when it receives valid traffic encryption key (TEK) from the key server (KS).On the other hand, the GM will set the stub-object 99 to state "DOWN" if it has no valid TEK because of errors, such as registration failure or TEK expiration before rekey.
Step 5	**exit** **Example:** `Device(config-gdoi-group)# exit`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Configuring GET VPN GM Authorization

GET VPN GM authorization can be done using preshared keys or PKI. It is a best practice to turn on GET VPN authorization. When a key server serves multiple GDOI groups, key server authorization is required to prevent a GM in one group from requesting keys and policies from another group. The ISAKMP authorization confirms that the GM is allowed to request GDOI attributes from the key server while the GDOI authorization confirms the GM is allowed to request GDOI attributes from a specific group configured in the key server.

To configure GET VPN GM authorization, perform either of the following tasks:

Configuring GM Authorization Using Preshared Keys

To configure GM authorization using preshared keys, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **server local**
5. **authorization address ipv4 {** *access-list-name* | *access-list-number* **}**

6. **exit**
7. **exit**
8. **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**time-range** *time-range-name*] [**fragments**] [**log** [*word*] | **log-input** [*word*]]
9. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group getvpn`	Identifies a GDOI and enters GDOI group configuration mode.
Step 4	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 5	**authorization address ipv4 {** *access-list-name* | *access-list-number* **}** **Example:** `Router(gdoi-local-server)# authorization address ipv4 50`	Specifies a list of addresses for a GDOI.
Step 6	**exit** **Example:** `Router(gdoi-local-server)# exit`	Exits GDOI local configuration mode and returns to GDOI group configuration mode.
Step 7	**exit** **Example:** `Router(config-gdoi-group)# exit`	Exits GDOI group configuration mode and returns to global configuration mode.
Step 8	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} *protocol source*	Defines an allowed IP access list.

	Command or Action	Purpose
	source-wildcard destination destination-wildcard [**precedence** *precedence*] [**tos** *tos*] [**time-range** *time-range-name*] [**fragments**] [**log** [*word*] | **log-input** [*word*]] **Example:** `Router(config)# access-list 50 permit ip 209.165.200.225 0.0.0.0 209.165.200.254 0.0.0.0`	• In the example, an access list with access list number 50 is defined, and packets sent from source IP address 209.165.200.225 to destination IP address 209.165.200.254 are permitted.
Step 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring GM Authorization Using PKI

To configure GM authorization using PKI, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp identity** {**address** | **dn** | **hostname**}
4. **crypto pki trustpoint** *name*
5. **subject-name** [*x.500-name*]
6. **exit**
7. **crypto gdoi group** *group-name*
8. **server local**
9. **authorization identity** *name*
10. **exit**
11. **exit**
12. **crypto identity** *name*
13. **dn** *name=string* [, *name=string*]
14. **exit**
15. **crypto isakmp identity** {**address** | **dn** | **hostname** }
16. **crypto pki trustpoint** *name*
17. **subject-name** [*x.500-name*]
18. **end**

DETAILED STEPS

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

	Command or Action	Purpose
	`Router> enable`	
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**crypto isakmp identity** {**address** | **dn** | **hostname**} **Example:** `Router(config)# crypto isakmp identity dn`	Defines the identity used by the router when the router is participating in the Internet Key Exchange (IKE) protocol.
Step 4	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint GETVPN`	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 5	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name OU=GETVPN`	Specifies the subject name in the certificate request.
Step 6	**exit** **Example:** `Router(ca-trustpoint)# exit`	Exits ca-trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto gdoi group** *group-name* **Example:** `Router(config)# crypto gdoi group getvpn`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 8	**server local** **Example:** `Router(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 9	**authorization identity** *name* **Example:** `Router(gdoi-local-server)# authorization identity GETVPN_FILTER`	Specifies an identity for a GDOI group.
Step 10	**exit** **Example:** `Router(gdoi-local-server)# exit`	Exits GDOI local server configuration mode and returns to GDOI group configuration mode.

	Command or Action	Purpose
Step 11	**exit** **Example:** `Router(config-gdoi-group)# exit`	Exits GDOI group configuration mode and returns to global configuration mode.
Step 12	**crypto identity** *name* **Example:** `Router(config)# crypto identity GETVPN_FILTER`	Configures the identity of the router with a given list of DNs in the certificate of the router and enters crypto identity configuration mode.
Step 13	**dn** *name=string* [, *name=string*] **Example:** `Router(config-crypto-identity)# dn ou=GETVPN`	Associates the identity of a router with the DN in the certificate of the router.
Step 14	**exit** **Example:** `Router(config-crypto-identity)# exit`	Exits GDOI group configuration mode and returns to global configuration mode.
Step 15	**crypto isakmp identity** {**address** | **dn** | **hostname** } **Example:** `Router(config)# crypto isakmp identity dn`	Defines the identity used by the router when the router is participating in the IKE protocol.
Step 16	**crypto pki trustpoint** *name* **Example:** `Router(config)# crypto pki trustpoint GETVPN`	Declares the trustpoint that your router should use and enters ca-trustpoint configuration mode.
Step 17	**subject-name** [*x.500-name*] **Example:** `Router(ca-trustpoint)# subject-name ou=getvpn`	Specifies the subject name in the certificate request.
Step 18	**end** **Example:** `Router(ca-trustpoint)# exit`	Exits GDOI group configuration mode, saves the configuration, and returns to privileged EXEC mode.

Verifying and Troubleshooting Cisco Group Encrypted Transport VPN Configurations

The following tasks can be used to verify and troubleshoot your GET VPN configurations. These tasks are optional and are used to gather information during troubleshooting.

Note With CSCsi82594, if Time-based Anti-Replay (TBAR) is enabled, the rekey time period is set to 2 hours (7200 seconds). In this scenario, the Key Server periodically sends a rekey to the Group Members every 2 hours (7200 seconds). In the below example, even though the Traffic Encryption Key (TEK) lifetime is set to 28800 seconds (8 hours), the rekey timer is still 2 hours. For show outputs displaying TBAR information, use the **show crypto gdoi gm replay** and **show crypto gdoi ks replay** commands.

```
crypto ipsec profile atm-profile
set security-association lifetime seconds 28800
!
crypto gdoi group ATM-DSL
server local
   sa ipsec 1
      !
      replay time window-size 100
```

Verifying Active Group Members on a Key Server

To verify active group members on a key server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi ks members**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi ks members** **Example:** `Router# show crypto gdoi ks members`	Displays information about key server members.

Verifying Rekey-Related Statistics

To verify rekey-related statistics, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi ks rekey**
3. **show crypto gdoi [gm]**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show crypto gdoi ks rekey**

Example:

```
Device# show crypto gdoi ks rekey

For group diffint (handle: 2147483650) server 10.0.8.1 (handle: 2147483650):

# of teks : 1 Seq num : 0
KEK POLICY (transport type : Unicast)
spi : 0xA8110DE7CC8B0FB201F2A8BFAA0F2D90
management alg : disabled encrypt alg : 3DES
crypto iv length : 8 key size : 24
orig life(sec): 300 remaining life(sec): 296 <------- ticking down
sig hash algorithm : enabled sig key length : 94
sig size : 64
sig key name : mykeys
```

On the key server, this command displays information about the rekeys that are being sent from the key server. The output displays the ticking down of the KEK remaining lifetime.

Step 3 **show crypto gdoi [gm]**

Example:

```
Device# show crypto gdoi
GROUP INFORMATION

Group Name : diffint
Group Identity : 3333
Rekeys received : 0
IPSec SA Direction : Both

Group Server list : 10.0.8.1

Group member : 10.0.3.1 vrf: None
Version : 1.0.2
Registration status : Registered
Registered with : 10.0.8.1
Re-registers in : 93 sec <-----------re-registration time for TEK or KEK
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 10.0.8.1:
access-list permit ip host 10.0.1.1 host 239.0.1.1
access-list permit ip host 10.0.100.2 host 238.0.1.1

KEK POLICY:
Rekey Transport Type : Unicast
Lifetime (secs) : 255 <-----------lifetime ticking
Encrypt Algorithm : 3DES
Key Size : 192
Sig Hash Algorithm : HMAC_AUTH_SHA
Sig Key Length (bits) : 512
```

On the group member, this command displays information about the rekeys that are being sent from the key server. The "re-registers in" field of the output displays the duration after which the group member re-registers for a TEK or a KEK, whichever time is smaller

Verifying IPsec SAs That Were Created by GDOI on a Group Member

To verify IPsec SAs that were created by GDOI on a group member, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi group** *group-name* **ipsec sa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi group** *group-name* **ipsec sa** **Example:** `Router# show crypto gdoi group diffint ipsec sa`	Displays information about IPsec SAs that were created by GDOI on a group member. • In this case, information will be displayed only for group “diffint.” • For information about IPsec SAs for all groups, omit the **group** keyword and *group-name* argument.

Verifying IPsec SAs That Were Created by GDOI on a Key Server

To verify IPsec SAs that were created by GDOI on a key server, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto ipsec sa**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto ipsec sa** **Example:** `Device# show crypto ipsec sa`	Displays the settings used by current SAs.

Verifying the TEKs that a Group Member Last Received from the Key Server

To verify the TEKs that a GM last received from the KS, perform the following steps on the GM:

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi** **Example:** `Router# show crypto gdoi`	Displays the current GDOI configuration and the policy that is downloaded from the KS. The TEKs are listed in the TEK POLICY section. Without enabling debugging, you can use this command to compare the TEKs that a GM actually last received with the TEKs downloaded from the KS to the IPsec control plane (which you can view using the **show crypto ipsec sa** command).

Verifying Cooperative Key Server States and Statistics

To verify cooperative key server states and statistics, perform the following steps, using one or both of the **debug** and **show** commands shown.

SUMMARY STEPS

1. **enable**
2. **debug crypto gdoi ks coop**
3. **show crypto gdoi group** *group-name* **ks coop** [**version**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**debug crypto gdoi ks coop** **Example:** `Router# debug crypto gdoi ks coop`	Displays information about a cooperative key server.
Step 3	**show crypto gdoi group** *group-name* **ks coop** [**version**] **Example:** `Router# show crypto gdoi group diffint ks coop version`	Displays information for the group "diffint" and version information about the cooperative key server.

Verifying Antireplay Pseudotime-Related Statistics

To verify antireplay pseudotime-related statistics, perform the following steps using one or all of the **clear**, **debug**, and **show** commands.

SUMMARY STEPS

1. **enable**
2. **clear crypto gdoi group** *group-name* **replay**
3. **debug crypto gdoi replay**
4. **show crypto gdoi group** *group-name*
5. **show crypto gdoi group** *group-name* **ks replay**

DETAILED STEPS

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

	Command or Action	Purpose
	`Router> enable`	
Step 2	**clear crypto gdoi group** *group-name* **replay** **Example:** `Router# clear crypto gdoi group diffint replay`	Clears the replay counters.
Step 3	**debug crypto gdoi replay** **Example:** `Router# debug crypto gdoi replay`	Displays information about the pseudotime stamp that is contained in a packet.
Step 4	**show crypto gdoi group** *group-name* **Example:** `Router# show crypto gdoi group diffint`	Displays information about the current pseudotime of the group member. • It also displays the different counts that are related to the antireplay for this group.
Step 5	**show crypto gdoi group** *group-name* **ks replay** **Example:** `Router# show crypto gdoi group diffint ks replay`	Displays information about the current pseudotime of the key server.

Verifying the Fail-Close Mode Status of a Crypto Map

To verify the fail-close mode status of a crypto map, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **show crypto map gdoi fail-close**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto map gdoi fail-close** **Example:** `Router# show crypto map gdoi fail-close`	Displays information about the status of the fail-close mode.

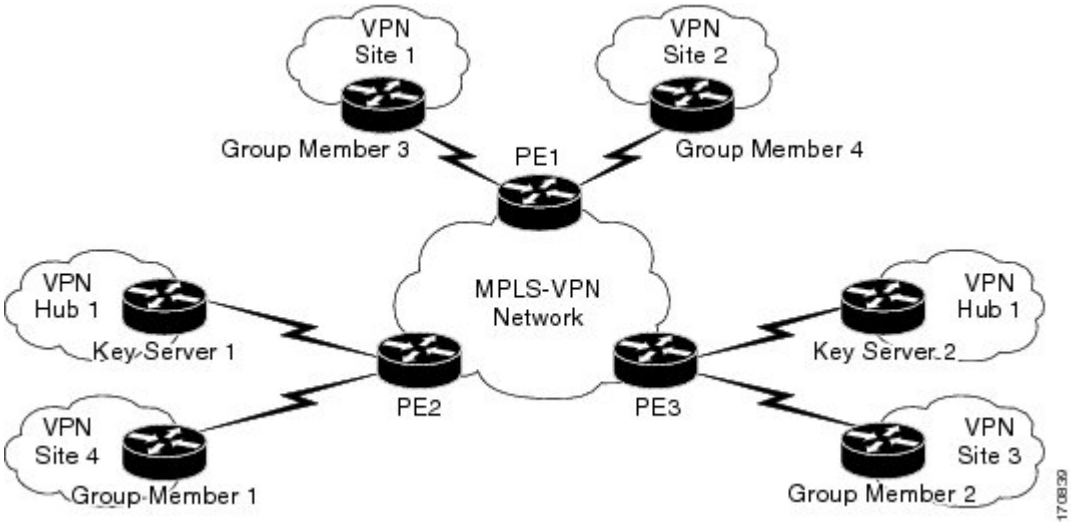
Configuration Examples for Cisco Group Encrypted Transport VPN

Example: Key Server and Group Member Case Study

The following case study includes encrypting traffic CE-CE in an MPLS VPN environment.

The MPLS VPN core interconnects VPN sites as is shown in the figure below. VPN site CPEs, Group Member 1 through Group Member 4, are grouped into a single GDOI group that correlates with a VPN with which these sites are a part. This scenario is an intranet VPN scenario. All the key servers and Group Members are part of the same VPN. Key Server 1 and Key Server 2 are the cooperative key servers that support VPN members Group Member 1 through Group Member 4. Key Server 1 is the primary key server and Key Server 2 is the secondary key server.

Figure 13: Key Server and Group Member Scenario



The following configuration examples are based on the case study in the figure above.

Example Key Server 1

Key server 1 is the primary key server.

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service internal
!
hostname KS1
!
logging buffered 100000 debugging
no logging console
!
no aaa new-model
```

```
!
resource policy
!
clock timezone EST 0
ip subnet-zero
no ip domain lookup
ip domain name cisco.com
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
 lifetime 400
crypto isakmp key cisco address 10.1.1.13
crypto isakmp key cisco address 10.1.1.9
crypto isakmp key cisco address 10.1.1.1
crypto isakmp key cisco address 10.1.1.5
crypto isakmp key cisco address 10.1.1.21
!
crypto ipsec transform-set gdoi-trans-group1 esp-3des esp-sha-hmac
!
crypto ipsec profile gdoi-profile-group1
 set security-association lifetime seconds 1800
 set transform-set gdoi-trans-group1
!
crypto gdoi group group1
 identity number 1
 server local
  rekey lifetime seconds 86400
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa group1-export-general
  rekey transport unicast
  sa ipsec 1
   profile gdoi-profile-group1
   match address ipv4 101
   replay counter window-size 64
  address ipv4 209.165.200.225
  redundancy
   local priority 10
   peer address ipv4 209.165.200.225
   !
interface Ethernet0/0
 ip address 209.165.200.225 255.255.255.252
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.18
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
end
```

Example Key Server 2

Key Server 2 is the secondary key server.

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
service internal
!
hostname KS2
!
```

```
logging buffered 100000 debugging
no logging console
!
no aaa new-model
!
resource policy
!
clock timezone EST 0
ip subnet-zero
no ip domain lookup
ip domain name cisco
!
crypto isakmp policy 1
 encr 3des
 authentication pre-share
 group 2
 lifetime 400
crypto isakmp key cisco address 10.1.1.9
crypto isakmp key cisco address 10.1.1.1
crypto isakmp key cisco address 10.1.1.5
crypto isakmp key cisco address 10.1.1.17
crypto isakmp key cisco address 10.1.1.13
!
crypto ipsec transform-set gdoi-trans-group1 esp-3des esp-sha-hmac
!
crypto ipsec profile gdoi-profile-group1
 set security-association lifetime seconds 1800
 set transform-set gdoi-trans-group1
!
crypto gdoi group group1
 identity number 1
 server local

  rekey lifetime seconds 86400
  rekey retransmit 10 number 2
  rekey authentication mypubkey rsa group1-export-general
  rekey transport unicast
  sa ipsec 1
   profile gdoi-profile-group1
   match address ipv4 101
   replay counter window-size 64
  address ipv4 10.1.1.21
  redundancy
   local priority 1
   peer address ipv4 10.1.1.17
   !
interface Ethernet0/0
 ip address 209.165.200.225 255.255.255.252
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.22
!
access-list 101 permit ip 10.1.0.0 0.0.255.255 10.1.0.0 0.0.255.255
!
end
```

Example: Configuring Group Member 1

Group member 1 is part of a GDOI group that correlates with a VPN with which these sites are a part.

```
service timestamps debug datetime msec
service timestamps log datetime msec
```

```
no service password-encryption
!
hostname GM1
!
resource policy
!
clock timezone EST 0
ip subnet-zero
!
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
 lifetime 3600
crypto isakmp key cisco address 209.165.200.225
crypto isakmp key cisco address 209.165.201.1
!
crypto gdoi group group1
 identity number 1
 server address ipv4 209.165.200.225
 server address ipv4 209.165.201.1
!
crypto map map-group1 10 gdoi
 set group group1
!
interface Ethernet0/0
 ip address 209.165.200.225 255.255.255.252
 crypto map map-group1
!
router bgp 1000
 no synchronization
 bgp log-neighbor-changes
 network 10.1.1.0 mask 255.255.255.0
 neighbor 10.1.1.2 remote-as 5000
 no auto-summary
!
ip classless
!
End
```

The same GDOI group cannot be applied to multiple interfaces. The following examples show unsupported cases:

Example 1

```
crypto map map-group1
  group g1
 interface ethernet 1/0
  crypto map map-group1
 interface ethernet 2/0
  crypto map map-group1
```

Example 2

```
crypto map map-group1 10 gdoi
  set group group1
crypto map map-group2 10 gdoi
  set group group1
 interface ethernet 1/0
  crypto map map-group1
 interface ethernet 2/0
```

Example: Configuring Group Member 2

Group member 2 is part of a GDOI group that correlates with a VPN with which these sites are a part.

```
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname GM2
!
resource policy
!
clock timezone EST 0
ip subnet-zero
!
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
 lifetime 3600
crypto isakmp key cisco address 209.165.200.225
crypto isakmp key cisco address 209.165.201.1
!
crypto gdoi group group1
 identity number 1
 server address ipv4 209.165.201.1
 server address ipv4 209.165.200.225
!
crypto map map-group1 10 gdoi
 set group group1
!
interface Ethernet0/0
 ip address 209.165.200.225 255.255.255.252
 crypto map map-group1
!
router bgp 2000
 no synchronization
 bgp log-neighbor-changes
 network 10.1.2.0 mask 255.255.255.0
 neighbor 10.1.1.6 remote-as 5000
 no auto-summary
!
ip classless
!
end
```

Example: Configuring Group Member 3

Group member 3 is part of a GDOI group that correlates with a VPN with which these sites are a part.

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname GM3
!
resource policy
!
clock timezone EST 0
ip subnet-zero
!
```

```
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
 lifetime 3600
crypto isakmp key cisco address 209.165.200.225
crypto isakmp key cisco address 209.165.201.1
!
crypto ipsec transform-set gdoi-trans-group1 esp-aes esp-sha-hmac
crypto gdoi group group1
 identity number 1
 server address ipv4 209.165.200.225
 server address ipv4 209.165.201.1
!
crypto map map-group1 10 gdoi
 set group group1
!
interface Ethernet0/0
 ip address 209.165.201.1 255.255.255.252
 crypto map map-group1
!
router bgp 3000
 no synchronization
 bgp log-neighbor-changes
 network 10.1.3.0 mask 255.255.255.0
 neighbor 10.1.1.10 remote-as 5000
 no auto-summary
!
ip classless
!
end
```

Example: Configuring Group Member 4

Group member 4 is part of a GDOI group that correlates with a VPN with which these sites are a part.

```
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname GM4
!
clock timezone EST 0
ip subnet-zero
!
crypto isakmp policy 1
 encr aes
 authentication pre-share
 group 14
 lifetime 3600
crypto isakmp key cisco address 209.165.200.225
crypto isakmp key cisco address 209.165.201.1
!
crypto gdoi group group1
 identity number 1
 server address ipv4 209.165.200.225
 server address ipv4 209.165.201.1
!
crypto map map-group1 10 gdoi
 set group group1
!
interface Ethernet0/0
```

```
 ip address 209.165.201.1 255.255.255.252
 crypto map map-group1
!
router bgp 4000
 no synchronization
 bgp log-neighbor-changes
 network 10.1.4.0 mask 255.255.255.0
 neighbor 10.1.1.14 remote-as 5000
 no auto-summary
!
ip classless
!
end
```

Example: Configuring Group Member 5

If a group member has multiple interfaces that are part of the same GDOI group, you should 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 key server.

The key server sees these as separate requests and must keep multiple records for the same group member, which also means sending multiple rekeys. If crypto is sourced from the loopback interface instead, the group member registers only once with the key server.

The following configuration shows how the group member registers once with the key server:

```
!


interface GigabitEthernet0/1
 description *** To AGG-1 ***
 crypto map dgvpn
!
interface GigabitEthernet0/2
 description *** To AGG-2 ***
 crypto map dgvpn
!
interface Loopback0
 ip address 209.165.201.1 255.255.255.255
!
 crypto map dgvpn local-address Loopback0
!
```

Example: Verifying the TEKs That a Group Member Last Received from the Key Server

The following example shows how to display the current GDOI configuration and the policy that is downloaded from the KS:

```
Device# show crypto gdoi

GROUP INFORMATION

    Group Name               : GETV6
.
.
.
KEK POLICY:
```

```
.
.
.
TEK POLICY for the current KS-Policy ACEs Downloaded:
  Ethernet2/0:
    IPsec SA:
        spi: 0x627E4B84(1652444036)
        transform: esp-aes
        sa timing:remaining key lifetime (sec): (3214)
        Anti-Replay(Time Based) : 10 sec interval
        tag method : cts sgt
        alg key size: 24 (bytes)
        sig key size: 20 (bytes)
        encaps: ENCAPS_TUNNEL


GROUP INFORMATION

    Group Name               : GETV4
.
.
.
KEK POLICY:
.
.
.
TEK POLICY for the current KS-Policy ACEs Downloaded:
  Ethernet2/0:
    IPsec SA:
        spi: 0xF6E6B597(4142314903)
        transform: esp-aes
        sa timing:remaining key lifetime (sec): (3214)
        Anti-Replay : Disabled
        tag method : cts sgt
        alg key size: 24 (bytes)
        sig key size: 20 (bytes)
        encaps: ENCAPS_TUNNEL
```

The TEKs are listed in the TEK POLICY section. Without enabling debugging, you can use this command to compare the TEKs that a GM actually last received with the TEKs downloaded from the KS to the IPsec control plane (which you can view using the **show crypto ipsec sa** command).

The tag method field shows the method used for GET VPN inline tagging; the possible values are either cts sgt (for Cisco TrustSec security group tags) or disabled. The alg key size field shows the key length for the encryption algorithm that is configured in the TEK policy. The sig key size field shows the key length for the signature that is configured in the TEK policy. The encaps field shows the type of IPsec encapsulation (either tunnel or transport) that is configured in the TEK policy.

The output from this command might show that a TEK has expired since the time it was received from the KS.

Example Passive SA

The following example displays information about crypto rules on outgoing packets:

```
Router# show crypto ruleset
Ethernet0/0:
  59 ANY ANY DENY
```

```
  11 ANY/848 ANY/848 DENY
 IP ANY ANY IPSec SA Passive
 IP ANY ANY IPSec Cryptomap
```

The following example displays the directional mode of the IPsec SA:

```
Router# show crypto ruleset detail
Ethernet0/0:
 20000001000019  59 ANY ANY DENY -> 20000001999999
20000001000029  11 ANY/848 ANY/848 DENY -> 20000001999999
 20000001000035 IP ANY ANY IPSec SA Passive
 20000001000039 IP ANY ANY IPSec Cryptomap
```

Example Fail-Close Mode

The following example shows that fail-close mode has been activated, and unencrypted traffic from access list 102 is allowed before the group member is registered:

```
crypto map map1 gdoi fail-close
 match address 102
 activate
crypto map map1 10 gdoi
 set group ks1_group
 match address 101
!
access-list 101 deny ip 10.0.1.0 0.0.0.255 10.0.1.0 0.0.0.255
access-list 102 deny tcp any eq telnet any
```

The following **show crypto map gdoi fail-close** command output shows that fail-close has been activated:

```
Router# show crypto map gdoi fail-close

Crypto Map: "svn"
        Activate: yes
        Fail-Close Access-List: (Deny = Forward In Clear, Permit = Drop)
            access-list 105 deny tcp any port = 23 any
            access-list 105 deny ospf any any
```

Additional References for Cisco Group Encrypted Transport VPN

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS security commands	*Cisco IOS Security Command References*

Related Topic	Document Title
Configuring IKE and IKE policy	"Configuring Internet Key Exchange for IPsec VPNs" module in the *Internet Key Exchange for IPsec VPNs Configuration Guide*
Configuring an IPsec transform	"Configuring Internet Key Exchange for IPsec VPNs" module in the *Internet Key Exchange for IPsec VPNs Configuration Guide*
Basic deployment guidelines for enabling GET VPN in an enterprise network	Cisco IOS GET VPN Solutions Deployment Guide
Routing control plane traffic (GDOI registrations and rekeys) through a separate VRF (for example, a dedicated management VRF)	Cisco IOS GETVPN VRF-Aware GDOI GM Solution Deployment Guide
Recommended cryptographic algorithms	Next Generation Encryption

Standards

Standard	Title
No new or modified standards are supported, and support for existing standards has not been modified.	—

MIBs

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

RFCs

RFC	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

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

Feature Information for Cisco Group Encrypted Transport VPN

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

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

Table 4: Feature Information for Cisco Group Encrypted Transport VPN

Feature Name	Releases	Feature Information
Cisco Group Encrypted Transport VPN	Cisco IOS XE Release 2.3	Cisco Group Encrypted Transport VPN is an optimal encryption solution for large-scale IP or MPLS sites that require any-to-any connectivity with minimum convergence time, low processing, provisioning, managing, and troubleshooting overhead. The following commands were introduced or modified: **address ipv4 (GDOI)**, **clear crypto gdoi**, **crypto gdoi gm**, **debug crypto gdoi**, **local priority**, **peer address ipv4**, **redundancy**, **rekey address ipv4**, **rekey transport unicast**, **replay counter window-size**, **replay time window-size**, **sa receive-only**, **show crypto gdoi**.
Create MIB Object to Track a Successful GDOI Registration	Cisco IOS XE Release 3.12S	The Create MIB Object to track a successful GDOI Registration feature introduces a new MIB object in the GDOI MIB to indicate the number of active TEKs in a group.

Feature Name	Releases	Feature Information
GET VPN Hardening	Cisco IOS XE Release 3.9S	This feature improves GET VPN resiliency. The improvements in resiliency prevent or minimize data-traffic disruption by using one of the following methods: • Making corrections when conditions that could cause a traffic disruption are detected. • Rapidly executing a recovery mechanism when a disruption is detected. The following commands were modified: **show crypto gdoi**, **show crypto ipsec sa**, **show tech-support**.
GET VPN IKEv1 Separation	Cisco IOS XE Release 3.11S	This feature eases maintenance and troubleshooting. The following commands were modified: **show tech-support**, **show crypto gdoi** and **show crypto ipsec sa**.
GET VPN Phase 1.2	Cisco IOS XE Release 2.3	These enhancements include the following features: • Change Key Server Role This feature enables you to change the role of the key server from primary to secondary. The following commands were added or modified for this feature: **clear crypto gdoi ks coop role** • Fail-Close Mode This feature prevents unencrypted traffic from passing through the group member before that member is registered. The following commands were added or modified for this feature: **activate**, **crypto map**, **match address**, and **show crypto map**. • Passive SA This feature allows a group member to be configured into passive mode permanently. The following command was introduced: **passive**.
GETVPN Policy-Change Enhancement for XE-based Group Members	Cisco IOS XE Fuji 16.8.1	The GETVPN Policy-Change Enhancement for XE-based Group Members feature enhances group members, running Cisco IOS XE software, handle policy change rekeys that require flow relocation. As a result of this feature, group members need not reregister and download again SAs and traffic that matches the old and new crypto policy is not leaked via clear text. This feature creates new inbound and outbound flows (SAs) for both old and new TEKs. No command was introduced or modified for this feature.

Feature Name	Releases	Feature Information
GETVPN Routing Awareness for BGP	Cisco IOS XE Release 3.13S	The GET VPN Routing Awareness for BGP 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 following commands were introduced or modified: **client status active-sa track**.
GET VPN Resiliency	Cisco IOS XE Release 3.9S	This feature improves the resiliency of GET VPN, so that data traffic disruption is prevented or minimized when errors occur. This feature introduces long SA lifetime functionality, which extends the maximum for which you can configure the lifetime of the key encryption key and traffic encryption keys from 24 hours to 30 days. This feature also lets you configure key servers to continue to send periodic reminder rekeys to group members that did not respond with an acknowledgment in the last scheduled rekey. By using a long SA lifetime in combination with periodic reminder rekeys, a key server can effectively synchronize group members if they miss a scheduled rekey before the keys roll over. The following commands were modified: **rekey lifetime**, **rekey retransmit**, **set security-association lifetime**, **show crypto gdoi**.
GET VPN Support of IPsec Inline Tagging for Cisco TrustSec	Cisco IOS XE Release 3.9S	Cisco TrustSec (CTS) uses the user and device identification information acquired during authentication to classify packets as they enter the network. CTS maintains classification of each packet by tagging packets with security group tags (SGTs) on ingress to the CTS network so that they 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 the access control policy based on the classification. The GET VPN Support of IPsec Inline Tagging for Cisco TrustSec feature uses GET VPN inline tagging to carry the SGT information across the private WAN. The following commands were introduced or modified: **show crypto gdoi**, **show crypto ipsec sa**, **tag cts sgt**.
GET VPN Time-Based Anti-Replay	Cisco IOS XE Release 2.3	Support for time-based antireplay was added to the Cisco VSA.

Feature Name	Releases	Feature Information
GET VPN Troubleshooting	Cisco IOS XE Release 3.8S	This feature provides improved debugging levels (so debug messages can be enabled per feature), event logging, exit trace capabilities to save a log of error conditions and their tracebacks, and conditional debugging (which provides the ability to debug individual group members from the key server). The conditional debugging feature provides the ability to perform conditional debugging on the key server so that it can filter based on GM or other cooperative key servers. The event logging feature provides the ability to log the last set of events. The following commands were introduced or modified: **clear crypto gdoi**, **debug crypto condition unmatched**, **debug crypto gdoi**, **debug crypto gdoi condition**, **monitor event-trace gdoi**, **show crypto gdoi**, and **show monitor event-trace gdoi**.
Group Encrypted Transport VPN Key Server	Cisco IOS XE Release 3.6S	Support was added for configuring a device running Cisco IOS XE as a key server. In Cisco IOS XE Release 3.6S, this feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers. In Cisco IOS XE Release 3.13S, support was added for the Cisco Cloud Services Router (CSR) 1000V Series.
VSA Support for GET VPN	Cisco IOS XE Release 2.3	Cisco VSA (high-performance crypto engine) support was added for GDOI and GET VPN.

Glossary

DOI—Domain of Interpretation. For Internet Security Association Key Management Protocol (ISAKMP), a value in the security association (SA) payload that describes in which context the key management message is being sent (IPsec or Group Domain of Interpretation).

GDOI—Group Domain of Interpretation. For ISAKMP, a means of distributing and managing keys for groups of mutually trusted systems.

group member—Device (Cisco IOS router) that registers with a group that is controlled by the key server for purposes of communicating with other group members.

group security association—SA that is shared by all group members in a group.

IPsec—IP security. Data encryption protocol for IP packets that are defined in a set of RFCs (see IETF RFC 2401).

ISAKMP—Internet Security Association and Key Management Protocol. Protocol that provides a framework for cryptographic key management protocols.

KEK—key encryption key. Key used to protect the rekey between the key server and group members.

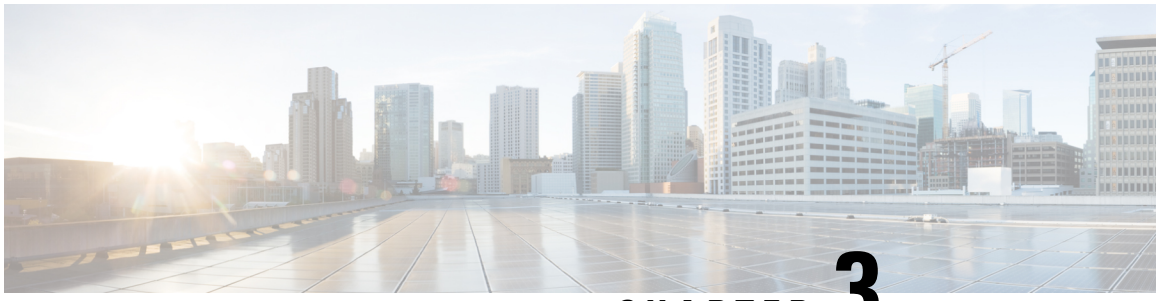
key server—Device (Cisco IOS router) that distributes keys and policies to group members.

MTU—maximum transmission unit. Size (in bytes) of the largest packet or frame that a given layer of a communications protocol can pass onward.

SA—security association. SA that is shared by all group members in a group.

Simple Network Management Protocol (SNMP)—An interoperable standards-based protocol that allows for external monitoring of a managed device through an SNMP agent.

TEK—traffic encryption key. Key that is used to protect the rekey between group members.



CHAPTER 3

GET VPN GM Removal and Policy Trigger

The GET VPN GM Removal and Policy Trigger feature lets you easily remove unwanted group members (GMs) from the group encrypted transport (GET) VPN network, provides a rekey triggering method to install new security associations (SAs) and remove obsolete SAs, and lets you check whether devices are running versions of GET VPN software that support these capabilities.

Finding Feature Information

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

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

Information About GM Removal and Policy Trigger

GET VPN Software Versioning

GET VPN software versions are of the form

major-version.minor-version.mini-version

where

- *major-version* defines compatibility for all GET VPN devices.
- *minor-version* defines compatibility for key server (KS)-to-KS (cooperative key server) associations and for GM-to-GM interoperability.

- *mini-version* tracks feature changes that have no compatibility impact.

For example, the base version (for all prior GET VPN features) is 1.0.1. Also, for example, the version that contains the GM removal feature and the policy replacement feature is 1.0.2, which means that these features are fully backward compatible with the base version (despite the introduction of behavior in these features for triggered rekeys).

GMs send the GET VPN software version to the KS in the vendor-ID payload during Internet Key Exchange (IKE) phase 1 negotiation (which is defined in RFC 2408, *Internet Security Association and Key Management Protocol [ISAKMP]*). KSs send the software version to other cooperative KSs in the version field of the cooperative KS announcement (ANN) messages. Cooperative KSs also synchronize their lists of versions that each GM is using.

The GM removal feature and the policy replacement feature each provide a command that you run on the KS (or primary KS) to find devices in the group that do not support that feature.

GM Removal

Without the GM removal and policy replacement features, you would need to complete the following steps to remove unwanted GMs from a group:

1. Revoke the phase 1 credential (for example, the preshared key or one or more PKI certificates).
2. Clear the traffic encryption key (TEK) and key encryption key (KEK) database on the KS.
3. Clear the TEK and KEK database on each GM individually and force each GM to re-register.

The third step is time-consuming when a GET VPN group serves thousands of GMs. Also, clearing the entire group in a production network might cause a network disruption. The GET VPN GM Removal and Policy Trigger feature automates this process by introducing a command that you enter on the KS (or primary KS) to create a new set of TEK and KEK keys and propagate them to the GMs.

GM Removal Compatibility with Other GET VPN Software Versions

You should use the GET VPN GM Removal and Policy Trigger feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature. Otherwise, secondary KSs or GMs running older software will ignore the GM removal message and continue to encrypt and decrypt traffic using the old SAs. This behavior causes network traffic disruption.

This feature provides a command that you use on the KS (or primary KS) to check whether all devices in the network are running versions that support GM removal. When the primary KS tries to remove GMs in a network containing devices that do not support GM removal, a warning message appears. For more information, see the “Ensuring That GMs Are Running Software Versions That Support GM Removal” section.

GM Removal with Transient IPsec SAs

The GET VPN GM Removal and Policy Trigger feature provides a command that you use on the KS (or primary KS) to trigger GM removal with transient IPsec SAs. This behavior shortens key lifetimes for all GMs and causes them to re-register before keys expire. During GM removal, no network disruption is expected, because traffic continues to be encrypted and decrypted using the transient IPsec SA until its lifetime expires. For more information, see the “Removing GMs with Transient IPsec SAs” section.

GM Removal with Immediate IPsec SA Deletion

The GET VPN GM Removal and Policy Trigger feature provides an optional keyword that you can use on the KS (or primary KS) to force GMs to delete old TEKs and KEKs immediately (without using transient SAs) and re-register. However, this behavior can cause a disruption to the data plane, so you should use this method only for important security reasons. For more information, see the "Removing GMs and Deleting IPsec SAs Immediately" section.

Policy Replacement and Rekey Triggering

The GET VPN GM Removal and Policy Trigger feature provides a new rekey triggering method to remove obsolete SAs and install new SAs.

Inconsistencies Regarding Which TEK and KEK Policy Changes Will Trigger Rekeys

Without this feature, there are inconsistencies regarding which TEK and KEK policy changes will trigger rekeys:

- Multiple rekeys could be sent during the course of security policy updates.
- Some policy changes (for example, transform set, profile, lifetime, and anti-replay) will install new SAs on GMs; however, the SAs from the existing policies remain active until their lifetimes expire.
- Some policy changes (for example, a TEK's access control entry/access control list (ACE/ACL) changes) will install new SAs on GMs and take effect immediately. However, the obsolete SAs are kept in each GM's database (and can be displayed using the **show crypto ipsec sa** command until their lifetimes expire).

For example, if the KS changes the policy from Data Encryption Standard (DES) to Advanced Encryption Standard (AES), when the GM receives this triggered rekey, it installs the new SAs (for example, for AES) and shortens the lifetimes of the old SAs (for example, for DES). The GM continues to encrypt and decrypt traffic using the old SAs until their shortened lifetimes expire.

Following is the formula to calculate the shortened lifetime:

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

where

- TEK_SLT is the TEK shortened lifetime
- TEK_RLT is the TEK remaining lifetime
- TEK_CLT is the TEK configured lifetime

The following table summarizes the inconsistencies regarding rekeys.

Table 5: Rekey Behavior After Security Policy Changes

Policy Changes	Rekey Sent?	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. Even if you enter the **clear crypto sa** command, it will re-register and download the old SA with the old lifetime again.
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 ACL. But the old SAs remain in the SA database (you can view them by using the **show crypto ipsec sa** command).
TEK: Enable replay counter	Yes	But 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	But the old SA with counter replay enabled remains active until its lifetime expires.
TEK: Enable TBAR	Yes	But the old SA with time-based anti-replay (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	But 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 behavior	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.

This feature solves these problems by ensuring consistency. With this feature, GET VPN policy changes alone will no longer trigger a rekey. When you change the policy (and exit 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 you then enter on the KS (or primary KS) to send a rekey (that is based on the latest security policy in the running configuration).

This feature also provides an extra keyword to the new command to force a GM receiving the rekey to remove the old TEKs and KEK immediately and install the new TEKs and KEK. Therefore, the new policy takes effect immediately without waiting for old policy SAs to expire. (However, using this keyword could cause a temporary traffic discontinuity, because all GMs might not receive the rekey message at the same time.)

Policy Replacement and Rekey Triggering Compatibility with Other GET VPN Software Versions

You should use rekey triggering only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature. For GMs running older versions that do not yet support the **crypto gdoi ks** command, the primary KS uses the software versioning feature to detect those versions and only triggers a rekey without sending instruction for policy replacement. Therefore, when a GM receives the rekey, it installs the new SAs but does not shorten the lifetimes of the old SAs. (This behavior is the same as the prior rekey method and ensures backward compatibility for devices that cannot support policy replacement.)

This feature provides a command that you use on the KS (or primary KS) to check whether all the devices in the network are running versions that support policy replacement. For more information, see the "Ensuring That GMs Are Running Software Versions That Support Policy Replacement" section.

How to Configure GET VPN GM Removal and Policy Trigger

Ensuring That GMs Are Running Software Versions That Support GM Removal

You should use the GET VPN GM Removal and Policy Trigger feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature. Otherwise, secondary KSs or GMs that are running older software will ignore the GM removal message and continue to encrypt and decrypt traffic using the old SAs. This behavior causes network traffic disruption.

Perform this task on the KS (or primary KS) to ensure that all devices in the network support GM removal.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature gm-removal**
3. **show crypto gdoi feature gm-removal | include No**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi feature gm-removal** **Example:** `Device# show crypto gdoi feature gm-removal`	Displays the version of the GET VPN software running on each KS and GM in the GET VPN network and displays whether that device supports GM removal.

	Command or Action	Purpose
Step 3	**show crypto gdoi feature gm-removal | include No** **Example:** `Device# show crypto gdoi feature gm-removal | include No`	(Optional) Displays only those devices that do not support GM removal.

Removing GMs with Transient IPsec SAs

Perform this task on the KS (or primary KS) to trigger removal of GMs with transient IPsec SAs.

SUMMARY STEPS

1. **enable**
2. **clear crypto gdoi** [**group** *group-name*] **ks members**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto gdoi** [**group** *group-name*] **ks members** **Example:** `Device# clear crypto gdoi ks members`	Creates a new set of TEK and KEK keys. This command also sends out GM removal messages to all GMs to clean up their old TEK and KEK databases.

Examples

A message appears on the KS as follows:

```
Device# 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...
```

After each GM receives the GM removal message, the following syslog message appears on each GM:

```
*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
```

Each GM removes the KEK immediately and shortens the lifetimes of the old TEKs as follows:

```
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
```

Also, the GMs start re-registering to the KS to obtain the new TEKs and KEK according to the conventional re-registration timer and with jitter (random delay) applied. Jitter prevents all GMs from reregistering at the same time and overloading the key server CPU. Only GMs that pass the authentication based on the new credential installed on the KS will receive the new TEKs and KEK.

GM removal should not cause a network disruption, because traffic continues to be encrypted and decrypted using the transient IPsec SA until its lifetime expires.

If you try to use this command on the secondary KS, it is rejected as follows:

```
Device# clear crypto gdoi ks members

ERROR for group GET: can only execute this command on Primary KS
```

Removing GMs and Deleting IPsec SAs Immediately

Perform this task on the KS (or primary KS) to force GMs to delete old TEKs and KEKs immediately and re-register.

SUMMARY STEPS

1. **enable**
2. **clear crypto gdoi** [**group** *group-name*] **ks members now**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**clear crypto gdoi** [**group** *group-name*] **ks members now** **Example:** `Device# clear crypto gdoi ks members now`	Creates a new set of TEK and KEK keys. This command also sends out GM removal messages to all GMs to clean up their old TEK and KEK databases. **Note** Using the **now** keyword can cause a network disruption to the data plane. Proceed with the GM removal only if a security concern is more important than a disruption.

Examples

A message appears on the KS as follows:

```
Device# 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...
```

After you enter the above command, the KS sends a “remove now” message to each GM to trigger the following actions on each GM:

1. Immediately cleans up its downloaded TEKs and KEK and its policy and returns to fail-open mode (unless fail-close mode is explicitly configured).
2. Sets up a timer with a randomly chosen period within 2 percent of the configured TEK lifetime.
3. When the timer in Step 2 expires, the GM starts re-registering to the KS to download the new TEKs and KEK.

On each GM, the following syslog message is displayed to indicate that the GM will re-register in a random time period:

```
*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
```

If you try to remove GMs in a network containing devices that do not support GM removal, a warning message appears:

```
Device# 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
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
```

Ensuring that GMs Are Running Software Versions That Support Policy Replacement

Perform this task on the KS (or primary KS) to check whether all devices in the network support policy replacement.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature policy-replace**
3. **show crypto gdoi feature policy-replace | include No**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**show crypto gdoi feature policy-replace** **Example:** `Device# show crypto gdoi feature policy-replace`	Displays the version of the GET VPN software running on each KS and GM in the GET VPN network and displays whether that device supports policy replacement.
Step 3	**show crypto gdoi feature policy-replace | include No** **Example:** `Device# show crypto gdoi feature policy-replace | include No`	(Optional) Finds only those devices that do not support policy replacement. For these devices, the primary KS sends only the triggered rekey without instructions for policy replacement. Therefore, when a GM receives the rekey, it installs the new SAs but does not shorten the lifetimes of the old SAs. This behavior is the same as the existing rekey method and ensures backward compatibility.

Triggering a Rekey

If you change the security policy (for example, from DES to AES) on the KS (or primary KS) and exit from global configuration mode, a syslog message appears on the KS indicating that the policy has changed and a rekey is needed. You enter the rekey triggering command as described below to send a rekey based on the latest policy in the running configuration.

Perform this task on the KS (or primary KS) to trigger a rekey.

SUMMARY STEPS

1. **enable**
2. **crypto gdoi ks** [**group** *group-name*] **rekey** [**replace-now**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto gdoi ks** [**group** *group-name*] **rekey** [**replace-now**] **Example:** `Device# crypto gdoi ks group mygroup rekey`	Triggers a rekey on all GMs. The optional **replace-now** keyword immediately replaces the old TEKs and KEK on each GM to enable the new policy before the SAs expire. **Note** Using the **replace-now** keyword could cause a temporary traffic discontinuity.

Examples

A message appears on the KS as follows:

```
Device# crypto gdoi ks rekey
Device#
*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
```

After the policy change, when each GM receives this triggered rekey, it installs the new SAs (for example, for AES) and shortens the lifetimes of the old SAs (for example, for DES). Each GM continues to encrypt and decrypt traffic using the old SA until its shortened lifetime expires.

If you try to trigger a rekey on the secondary KS, it rejects the command as shown below:

```
Device# crypto gdoi ks rekey
ERROR for group GET: This command must be executed on Pri-KS
```

Configuration Examples for GET VPN GM Removal and Policy Trigger

Example: Removing GMs from the GET VPN Network

Ensuring That GMs Are Running Software Versions That Support GM Removal

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to check whether all the devices in the network support the GM removal feature:

```
Device# show crypto gdoi feature gm-removal

Group Name: GET
 Key Server ID       Version   Feature Supported
    10.0.8.1         1.0.2          Yes
    10.0.9.1         1.0.2          Yes
    10.0.10.1        1.0.2          Yes
    10.0.11.1        1.0.2          Yes
 Group Member ID     Version   Feature Supported
    10.0.0.2         1.0.2          Yes
    10.0.0.3         1.0.1          No
```

The following example shows how to find only those devices that do not support GM removal:

```
Device# show crypto gdoi feature gm-removal | include No

       10.0.0.3          1.0.1          No
```

The above example shows that the GM with IP address 10.0.0.3 is running older software version 1.0.1 (which does not support GM removal) and should be upgraded.

Removing GMs with Transient IPsec SAs

The following example shows how to trigger GM removal with transient IPsec SAs. You use this command on the KS (or primary KS).

```
Device# 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...
```

Removing GMs and Deleting IPsec SAs Immediately

The following example shows how to force GMs to delete old TEKs and KEKs immediately and re-register. You use this command on the KS (or primary KS).

```
Device# 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...
```

Example: Triggering Rekeys on Group Members

Ensuring That GMs Are Running Software Versions That Support Rekey Triggering

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to display the version of software on devices in the GET VPN network and display whether they support rekey triggering after a policy change:

```
Device# show crypto gdoi feature policy-replace

Key Server ID       Version  Feature Supported
  10.0.8.1         1.0.2          Yes
  10.0.9.1         1.0.2          Yes
  10.0.10.1        1.0.2          Yes
  10.0.11.1        1.0.2          Yes
 Group Member ID     Version  Feature Supported
  5.0.0.2          1.0.2          Yes
  9.0.0.2          1.0.1          No
```

The following example shows how to find only those devices that do not support rekey triggering after policy replacement:

```
Device# show crypto gdoi feature policy-replace  | include No

       9.0.0.2          1.0.1          No
```

For these devices, the primary KS sends only the triggered rekey without instructions for policy replacement. Therefore, when a GM receives the rekey, it installs the new SAs but does not shorten the lifetimes of the old SAs.

Triggering a Rekey

The following example shows how to trigger a rekey after you have performed a policy change. In this example, an IPsec policy change (for example, DES to AES) occurs with the **profile gdoi-p2** command:

```
Device# configure terminal
Device(config)# crypto gdoi group GET
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# no profile gdoi-p
Device(gdoi-sa-ipsec)# profile gdoi-p2
Device(gdoi-sa-ipsec)# end
Device#

*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
Device# crypto gdoi ks rekey
Device#
*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
```

The following example shows the error message that appears if you try to trigger a rekey on the secondary KS:

```
Device# crypto gdoi ks rekey

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

Note If time-based antireplay (TBAR) is set, the key server periodically sends a rekey to the group members every 2 hours (7200 sec). In the following example, even though the lifetime is set to 8 hours (28800 sec), the rekey timer is set to 2 hours.

```
Device(config)# crypto ipsec profile atm-profile
Device(ipsec-profile)# set security-association lifetime seconds 28800
!
Device(ipsec-profile)# exit
Device(config)# crypto gdoi group ATM-DSL
Device(config-gdoi-group)# server local
Device(gdoi-sa-ipsec)# sa ipsec 1
!
Device(gdoi-sa-ipsec)# replay time window-size 100
```

The commands **show crypto gdoi gm replay** and **show crypto gdoi ks replay** displays TBAR information.

Additional References for GET VPN GM Removal and Policy Trigger

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS security commands	*Cisco IOS Security Command References*

Technical Assistance

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

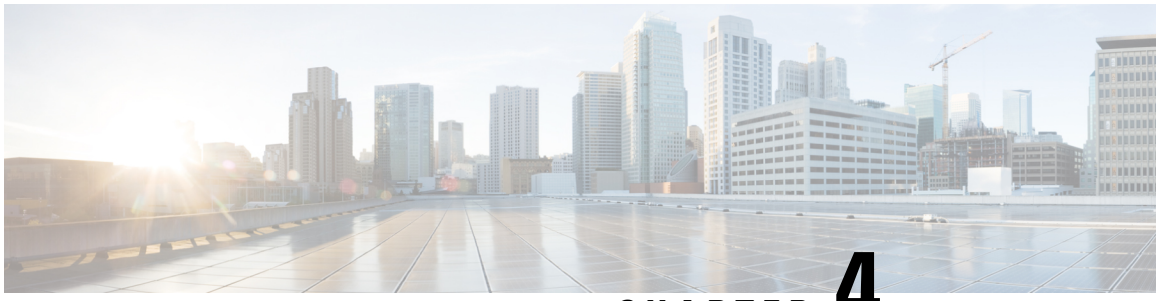
Feature Information for GET VPN GM Removal and Policy Trigger

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

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

Table 6: Feature Information for GET VPN GM Removal and Policy Trigger

Feature Name	Releases	Feature Information
GET VPN GM Removal and Policy Trigger	Cisco IOS XE Release 3.8S	This feature provides a command that lets you efficiently eliminate unwanted GMs from the GET VPN network, provides a rekey triggering command to install new SAs and remove obsolete SAs, and provides commands that display whether devices on the network are running versions of GET VPN software that support these features. The following commands were introduced or modified: **clear crypto gdoi, crypto gdoi ks, show crypto gdoi.**



CHAPTER 4

GDOI MIB Support for GET VPN

The existing MIBs in crypto are the Internet Key Exchange (IKE) and IP security (IPsec) MIBs, which are not sufficient for Group Domain of Interpretation (GDOI). The GDOI MIB Support for GET VPN feature adds MIB support for RFC 6407, The Group Domain of Interpretation ; it supports only the objects related to the GDOI MIB IETF standard. You can import the GDOI MIB .my file into an SNMP management station and parse it to retrieve the table objects and hierarchy information.

The GDOI MIB consists of objects and notifications (formerly called traps) that include information about GDOI groups, group menber (GM) and key server (KS) peers, and the policies that are created or downloaded. Only "get" operations are supported for GDOI.

To configure GDOI MIB support for GET VPN, see the "Configuring GDOI MIB Support for GET VPN" section.

Finding Feature Information

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

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

Information About GDOI MIB Support for GET VPN

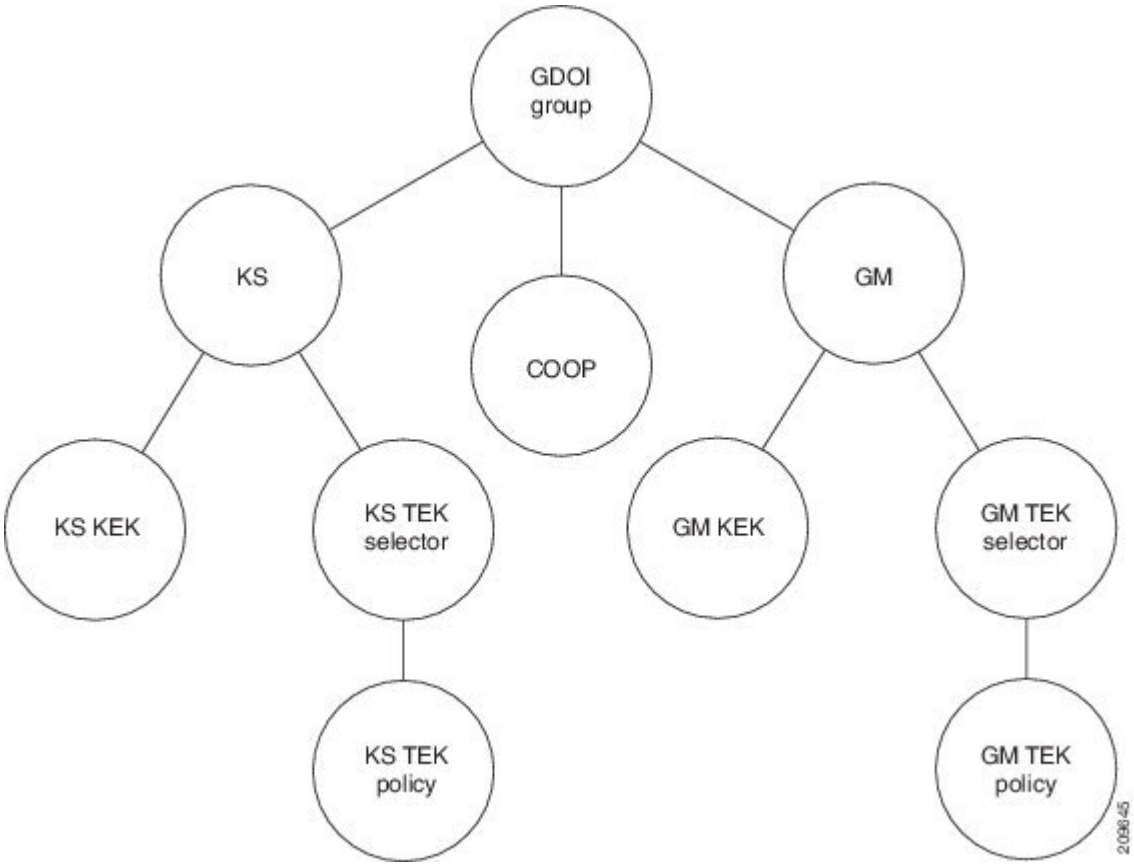
GDOI MIB Compatibility with Other GET VPN Software Versions

The GDOI MIB Support for GET VPN feature provides a command that you use on the KS (or primary KS) to check whether all the devices in the network are running versions that support the GDOI MIB. For more information, see the "Ensuring that GMs Are Running Software Versions That Support the GDOI MIB" section.

GDOI MIB Table Hierarchy

The GDOI MIB objects are organized into the following GDOI MIB tables. Following is the relationship (hierarchy) among the tables:

Figure 14: GDOI MIB Table 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.
- Group member count -- Specifies the number of registered KSs to this group.
- Group active peer KS count -- Specifies the number of active KSs to this group.
- Group last rekey retransmits -- Specifies the cumulative count of number of rekey messages and retransmit messages sent as a part of last rekey operation.
- Group last rekey time taken -- Specifies the time taken by the KS to complete the last rekey operation.

KS table objects:

- KS ID type
- KS ID length
- KS ID value
- Active KEK—SPI of the key encryption key (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.
- KS Role -- Primary or secondary.
- Number of registered GMs -- count of GMs registered to this KS.

COOP table objects:

- COOP peer ID type
- COOP peer ID length
- COOP peer ID value
- COOP peer ID role -- Primary or secondary
- COOP peer status -- Alive, dead or unknown
- Number of registered GMs -- count of GMs registered to the COOP peer

GM table:

- 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.
- Count of active TEKs -- number of active TEKs used by the GM to encrypt/decrypt/authenticate dataplane traffic.

KS KEK table:

- 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 (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 for traffic encryption keys (TEK).
- 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 6407).

KS TEK policy table:

- 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:

- 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 (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 6407).

GM TEK policy table:

- 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 Notifications

The GDOI MIB supports the Simple Network Management Protocol (SNMP) notifications in the following table. The GDOI MIB contains two kinds of notifications: those generated by the KS and those generated by each GM. You can enable any combination of notifications (or all notifications).

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 reregistration 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.

For more information, see the “Enabling GDOI MIB Notifications” section.

GDOI MIB Limitations

The GDOI MIB contains only objects that are listed in RFC 6407 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 GET VPN-specific features

How to Configure GDOI MIB Support for GET VPN

Ensuring that GMs Are Running Software Versions That Support the GDOI MIB

Perform this task on the KS (or primary KS) to ensure that all devices in the GET VPN network support the GDOI MIB.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature gdoi-mib**
3. **show crypto gdoi feature gdoi-mib | include No**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi feature gdoi-mib** **Example:** `Device# show crypto gdoi feature gdoi-mib`	Displays the version of the GET VPN software running on each KS and GM in the network and displays whether that device supports the GDOI MIB.
Step 3	**show crypto gdoi feature gdoi-mib | include No** **Example:** `Device# show crypto gdoi feature gdoi-mib | include No`	(Optional) Finds only those devices that do not support the GDOI MIB.

Creating Access Control for an SNMP Community

You specify an SNMP community access string to define the relationship between the SNMP manager and the SNMP agent on the KS or GM in order to permit access to SNMP. Your community access string acts like a password to regulate access to the agent on the device.

Perform this task to specify the community access string.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server community** *community-string* [**view** *view-name*] [**ro** | **rw**] [**ipv6** *nacl*] [*access-list-number* | *extended-access-list-number* | *access-list-name*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server community** *community-string* [**view** *view-name*] [**ro**	**rw**] [**ipv6** *nacl*] [*access-list-number*	*extended-access-list-number*	*access-list-name*] **Example:** `Device(config)# snmp-server community mycommunity`	Specifies the community access string.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode, saves the configuration, and returns to privileged EXEC mode.

For more information about specifying a community access string, refer to the "Configuring SNMP Support" module in the *SNMP Configuration Guide*. For more information about the **snmp-server community** command (including syntax and usage guidelines), refer to the Cisco IOS SNMP Support Command Reference.

Enabling Communication with the SNMP Manager

Perform this task to enable communication between the SNMP agent on the KS or GM and the SNMP manager.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server host** {*hostname* | *ip-address*} **version** {**1** | **2c** | **3**} *community-string*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server host** {*hostname* | *ip-address*} **version** {**1** | **2c** | **3**} *community-string* **Example:** `Device(config)# snmp-server host 209.165.200.225 version 2c mycommunity`	Specifies the host to receive SNMP notifications. • 2c is usually used as the SNMP version.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode, saves the configuration, and returns to privileged EXEC mode.

For more information about enabling communication with the SNMP manager, refer to the "Configuring SNMP Support" module in the *SNMP Configuration Guide*. For more information about the **snmp-server host** command (including syntax and usage guidelines), refer to the Cisco IOS SNMP Support Command Reference.

Enabling GDOI MIB Notifications

Perform this task to enable GDOI MIB notifications on the KS or GM.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps gdoi** [*notification-type*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps gdoi** [*notification-type*] **Example:** `Device(config)# snmp-server enable traps gdoi gm-registration-complete gm-rekey-rcvd ks-new-registration ks-reg-complete`	Specifies the particular SNMP notifications to be enabled. You can specify any combination of the following types in any order. If you enter the command without any of the following keywords, all GDOI MIB notifications are enabled. • **gm-incomplete-cfg**—A GM sent an error notification because of a missing configuration. • **gm-re-register**—A GM began the reregistration process with a KS. • **gm-registration-complete**—A GM completed registration to a KS. • **gm-rekey-fail**—A GM sent an error notification because it cannot successfully process and install a rekey. • **gm-rekey-rcvd**—A rekey message was received by a GM. • **gm-start-registration**—A GM first sent a registration request to a KS. • **ks-new-registration**—A KS first received a registration request from a GM. • **ks-no-rsa-keys**—An error notification was received from the KS because of missing RSA keys. • **ks-reg-complete**—A GM completed registration to the KS. • **ks-rekey-pushed**—A rekey message was sent by the KS. • **ks-gm-deleted**—A GM is deleted by the KS. • **ks-peer-reachable**—An unreachable COOP peer becomes reachable.

	Command or Action	Purpose
		• **ks-peer-unreachable**—A reachable COOP peer becomes unreachable. • **ks-role-change**—A KS changes it's role from primary to secondary or vice-versa.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode, saves the configuration, and returns to privileged EXEC mode.

Configuration Examples for GDOI MIB Support for GET VPN

Example: Ensuring That GMs Are Running Software Versions That Support the GDOI MIB

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to check whether all the devices in the network support the GDOI MIB:

```
Device# show crypto gdoi feature gdoi-mib

Group Name: GET
 Key Server ID       Version   Feature Supported
    10.0.8.1         1.0.2          Yes
    10.0.9.1         1.0.2          Yes
    10.0.10.1        1.0.2          Yes
    10.0.11.1        1.0.2          Yes
 Group Member ID     Version   Feature Supported
    10.0.11.2          1.0.2          Yes
    10.0.11.3          1.0.1          No
```

The following example shows how to find only those devices that do not support the GDOI MIB:

```
Device# show crypto gdoi feature gdoi-mib | include No

       10.0.11.3          1.0.1          No
```

Example: Creating Access Control for an SNMP Community

The following example shows how to specify an SNMP community string named mycommunity to define the relationship between the SNMP manager and the SNMP agent on the KS or GM in order to permit access to SNMP:

```
Device> enable
```

```
Device# configure terminal
Device(config)# snmp-server community mycommunity
Device(config)# end
```

Example: Enabling Communication with the SNMP Manager

The following example shows how to enable communication with the SNMP manager. This example using a community string named mycommunity that has already been created:

```
Device> enable
Device# configure terminal
Device(config)# snmp-server host 209.165.200.225 version 2c mycommunity
Device(config)# end
```

Example: Enabling GDOI MIB Notifications

The following example shows how to enable GDOI MIB notifications:

```
Device> enable
Device# configure terminal
Device(config)# snmp-server enable traps gdoi gm-registration-complete gm-rekey-rcvd
ks-new-registration ks-reg-complete
Device(config)# end
```

Additional References for GDOI MIB Support for GET VPN

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS security commands	*Cisco IOS Security Command References*
Configuring SNMP	• "Configuring SNMP Support" module in the SNMP Configuration Guide, Cisco IOS Release 15.2M&T • Cisco IOS SNMP Support Command Reference

MIBs

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

Technical Assistance

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

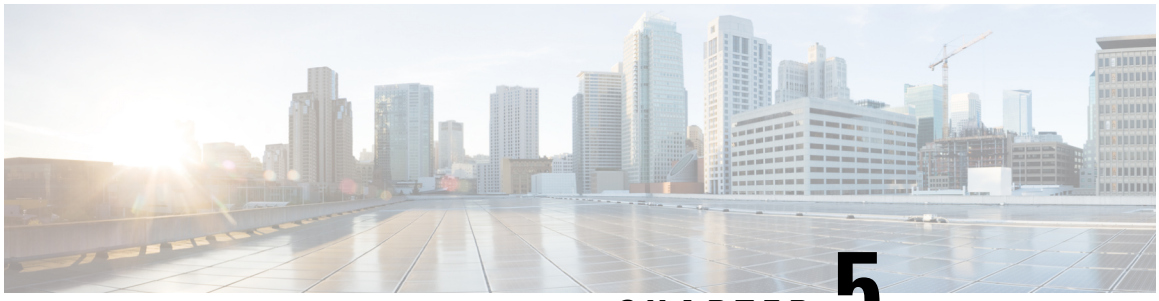
Feature Information for GDOI MIB Support for GET VPN

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

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

Table 8: Feature Information for GDOI MIB Support for GET VPN

Feature Name	Releases	Feature Information
GDOI MIB Support for GET VPN	Cisco IOS XE Release 3.8S	This feature adds MIB support for IETF RFC 6407, The Group Domain of Interpretation. This feature supports only the objects related to the GDOI MIB IETF standard. This feature also provides a command that displays whether devices on the network are running versions of GET VPN software that support the GDOI MIB. The GDOI MIB consists of objects and notifications that include information about GDOI groups, GM and KS peers, as well as the policies that are created or downloaded. The following command was introduced: **snmp-server enable traps gdoi.**
XE 3.16 GETVPN GDOI/COOP MIBS	Cisco IOS XE Release 3.16S	The XE 3.16 GETVPN GDOI/COOP MIBS feature enhances the existing CISCO-GDOI-MIB via additional SNMP MIB objects for key server parameters and traps for key server events. The following command was modified: **snmp-server enable traps gdoi.**



CHAPTER 5

GET VPN Resiliency

The GET VPN Resiliency feature improves the resiliency of Cisco Group Encrypted Transport (GET) VPN so that data traffic disruption is prevented or minimized when errors occur.

Finding Feature Information

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

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

Prerequisites for GET VPN Resiliency

All key servers (KSs) and group members (GMs) on which you want to enable this feature must be running GET VPN software version 1.0.4 or higher. You should use this feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature. This feature provides a command that you use on the KS (or primary KS) to check whether all devices in the network are running versions that support this feature. For more information, see the "*Ensuring That GMs Are Running Software Versions That Support Long SA Lifetime*" section.

Restrictions for GET VPN Resiliency

- All key servers (KSs) and group members (GMs) must be upgraded for Long SA Lifetime.

Information About GET VPN Resiliency

Long SA Lifetime

The long security association (SA) lifetime functionality extends the maximum lifetime of the key encryption key (KEK) and traffic encryption key (TEK) from 24 hours to 30 days. From Cisco IOS XE Everest 16.6, for a KEK or TEK, a lifetime of 24 hours or longer is considered a long SA lifetime. This functionality also lets you configure key servers (KSs) to continue to send periodic reminder rekeys to group members (GMs) that do not respond with an acknowledgment in the last scheduled rekey.

By using a long SA lifetime in combination with periodic reminder rekeys, a KS can effectively synchronize GMs if they miss a scheduled rekey before the keys roll over.

Note For a lifetime of 24 hours or longer, the encryption algorithm must be Advanced Encryption Standard-cipher block chaining (AES-CBC) or Advanced Encryption Standard-Galois/Counter Mode (AES-GCM) with an AES key of 128 bits or stronger.

You can use the long SA lifetime functionality along with the GETVPN Suite-B feature to use AES-GSM and Galois Message Authentication Code-Advanced Encryption Standard (GMAC-AES) as traffic encryption key (TEK) policy transforms in a group for packets encapsulated with GCM-AES and GMAC-AES.

Migrating to Long SA Lifetime

When migrating to the long SA lifetime functionality (greater than or equal to one day), the following rules apply:

- When a long SA lifetime is configured on a crypto IPsec profile, GETVPN displays a warning message to not use the IPsec profile for a non- Group Domain of Interpretation (GDOI) group.
- If group members are registered to a key server with short SA lifetime and the key server changes the policy to long SA lifetime, GETVPN checks the software version of all the GMs when the **crypto gdoi ks rekey** command is configured to initiate the policy change. If the GMs registered with the KS do not support long SA lifetime, a message is displayed to discourage the policy change until all GMs are upgraded.
- When the Long SA feature is enabled in KS, it will block registration from GMs running older Cisco IOS releases, which does not support this feature.
- When the lifetime of KEK or TEK is a long SA lifetime set to 24 hours (86400 seconds) or longer, the rekey lifetime is set to half the KEK or TEK lifetime.

Clock Skew Mitigation

Sometimes with longer security association (SA) lifetimes, a group member (GM) may not receive updates from a key server for a longer duration. This may result in group members experiencing clock skew for key encryption key (KEK) lifetime, traffic encryption key (TEK) lifetime, and Time-Based Anti-Replay (TBAR) pseudotime. The refresh rekey and rollover to new outbound IPsec SA helps GMs in mitigating clock skew issues.

Refresh Rekey

If the traffic encryption key (TEK) lifetime is set for a duration greater than two days and Time-Based Anti-Replay (TBAR) is disabled, a key server sends a refresh rekey every 24 hours which updates the key encryption key (KEK) lifetime, TEK lifetime, and TBAR pseudotime on all group members (GMs). In simple terms, a refresh rekey is a retransmission of the current KEK policy, TEK policy, and TBAR pseudotime (if enabled) to all GMs, regardless of the status of receiving a unicast acknowledgment (ACK) for the last rekey. If TBAR is enabled, the refresh rekey is sent every two hours to synchronize the pseudotime, so that an additional refresh rekey is not required.

Rollover to New Outbound IPsec SA

When a long SA lifetime (greater than or equal to one day) is configured, the rollover happens when the remaining lifetime of the traffic encryption key (TEK) reaches 1% of the old TEK configured lifetime that has a lower limit of 30 seconds and not 30 seconds of the old TEK's remaining lifetime. This allows a greater clock skew between the group members (GMs) before discarding traffic from one GM rolling over to the new TEK late (after the other GM has already deleted the old TEK). This mitigates the GM from being "offline" (disconnected from the KS) for a long duration and from being unable to receive the refresh rekeys to mitigate the clock skew.

Periodic Reminder Sync-Up Rekey

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.

Pre-Positioned Rekey

The pre-positioned rekey functionality allows the key server (KS) to send a rekey earlier than half the duration of the SA lifetime, when a longer SA lifetime (greater than or equal to one day) is configured. The normal behavior of sending the rekey is used for a short SA lifetime. When group members (GMs) receive this early rekey, they continue to use the old TEK as outbound until rolled over to the new TEK as outbound. The pre-positioned rekey feature along with the Long SA Lifetime feature improves key rollover stability. This functionality allows the (KS) sufficient time to recover rekey errors, such as periodic reminder rekeys and synchronize rekeys.

How to Configure GET VPN Resiliency

Ensuring That GMs Are Running Software Versions That Support Long SA Lifetime

You should use the Long SA Lifetime feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature.

Perform this task on the key server (or primary key server) to ensure that all devices in the network support long SA lifetime.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature long-sa-lifetime**
3. **show crypto gdoi feature long-sa-lifetime | include No**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi feature long-sa-lifetime** **Example:** `Device# show crypto gdoi feature long-sa-lifetime`	Displays the version of the GET VPN software running on each KS and GM in the GET VPN network and displays whether that device supports long SA lifetime.
Step 3	**show crypto gdoi feature long-sa-lifetime | include No** **Example:** `Device# show crypto gdoi feature long-sa-lifetime | include No`	(Optional) Displays only those devices that do not support long SA lifetime.

Configuring Long SA Lifetime

Configuring Long SA Lifetime for TEK

To configure long SA lifetime for traffic encryption key (TEK), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec profile** *name*
4. **set security-association lifetime days** *days*

5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec profile** *name* **Example:** `Device(config)# crypto ipsec profile gdoi-p`	Defines the IPsec parameters that are to be used for IPsec encryption between two IPsec devices and enters crypto IPsec profile configuration mode.
Step 4	**set security-association lifetime days** *days* **Example:** `Device(ipsec-profile)# set security-association lifetime days 15`	Configures the security association (SA) lifetime to one day or longer. • The maximum number of days is 30.
Step 5	**end** **Example:** `Device(ipsec-profile)# end`	Exits crypto IPsec profile configuration mode and returns to privileged EXEC mode.

Configuring Long SA Lifetime for KEK

To configure long SA lifetime for key encryption key (TEK), perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **identity number** *number*
5. **server local**
6. **rekey lifetime days** *days*
7. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GET`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**identity number** *number* **Example:** `Device(config-gdoi-group)# identity number 3333`	Identifies a GDOI group number.
Step 5	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**rekey lifetime days** *days* **Example:** `Device(gdoi-local-server)# rekey lifetime days 20`	Limits the number of days or seconds for a KEK.
Step 7	**end** **Example:** `Device(gdoi-local-server)# end`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.

Configuring the Periodic Reminder Sync-Up Rekey

To configure the periodic reminder sync-up rekey, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **identity number** *number*
5. **server local**
6. **rekey retransmit** *number-of-seconds* **periodic**
7. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group group1`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**identity number** *number* **Example:** `Device(config-gdoi-group)# identity number 3333`	Identifies a GDOI group number.
Step 5	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 6	**rekey retransmit** *number-of-seconds* **periodic** **Example:** `Device(gdoi-local-server)# rekey retransmit 10 periodic`	Specifies the number of times the rekey message is periodically retransmitted. • If this command is not configured, there will be no retransmits.
Step 7	**end** **Example:** `Device(gdoi-local-server)# end`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.

Verifying and Troubleshooting GET VPN Resiliency

Verifying and Troubleshooting GET VPN Resiliency on a Key Server

To view the configuration that is running on a key server (KS), use the **show running-config** command and the following commands.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi**
3. **show crypto gdoi ks rekey**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**show crypto gdoi** **Example:** `Device# show crypto gdoi`	Displays the current GDOI configuration and the policy that is downloaded from the KS.
Step 3	**show crypto gdoi ks rekey** **Example:** `Device#  show crypto gdoi ks rekey`	Displays information about the rekeys that are sent from the KS.

Verifying and Troubleshooting GET VPN Resiliency on a Group Member

To view the configuration that is running on a group member (GM), use the **show running-config** command and the following commands.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi ks rekey**
3. **show crypto gdoi ks policy**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi ks rekey** **Example:** `Device#  show crypto gdoi ks rekey`	Displays information about the rekeys that are sent from the KS.
Step 3	**show crypto gdoi ks policy** **Example:** `Device# show crypto gdoi ks policy`	Displays the time until the next rekey.

Configuration Examples for GET VPN Resiliency

Example: Ensuring That GMs Are Running Software Versions That Support Long SA Lifetime

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to check whether all the devices in each group support long SA lifetimes:

```
Device# show crypto gdoi feature long-sa-lifetime

Group Name: GETVPN
    Key Server ID       Version   Feature Supported
        10.0.5.2         1.0.4          Yes
        10.0.6.2         1.0.4          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         1.0.4          Yes
        10.0.3.2         1.0.4          Yes
```

You can also enter the above command on a GM (which will display 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 GET VPN network that do *not* support long SA lifetimes:

```
Device# show crypto gdoi feature long-sa-lifetime | 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
```

Example: Configuring Long SA Lifetime

Example: Configuring Long SA Lifetime for TEK

The following example shows how to configure the long SA lifetime for traffic encryption key (TEK):

```
Device> enable
Device# configure terminal
Device(config)# crypto ipsec profile gdoi-p
Device(ipsec-profile)# set security-association lifetime days 15
Device(ipsec-profile)# end
```

Example: Configuring Long SA Lifetime for KEK

The following example shows how to configure the long SA lifetime for key encryption key (KEK):

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group GET
Device(config-gdoi-group)# identity number 3333
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# rekey lifetime days 20
Device(gdoi-local-server)# end
```

Example: Configuring the Periodic Reminder Sync-Up Rekey

The following example shows how to configure the periodic reminder sync-up rekey:

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group group1
Device(config-gdoi-group)# identity number 3333
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# rekey retransmit 10 periodic
Device(gdoi-local-server)# end
```

Additional References for GET VPN Resiliency

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Cisco IOS security commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*
Basic deployment guidelines for enabling GET VPN in an enterprise network	*Cisco IOS GET VPN Solutions Deployment Guide*
Designing and implementing a GET VPN network	*Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

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

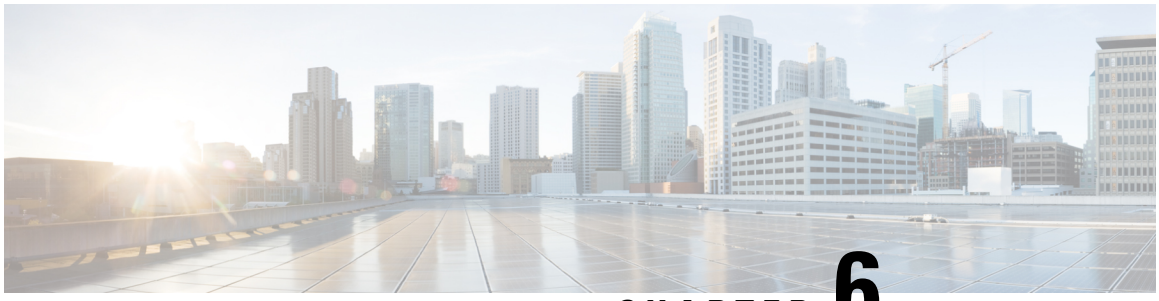
Feature Information for GET VPN Resiliency

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

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

Table 9: Feature Information for GET VPN Resiliency

Feature Name	Releases	Feature Information
GET VPN Resiliency	Cisco IOS XE Release 3.9S	The GET VPN Resiliency feature improves the resiliency of Cisco Group Encrypted Transport (GET) VPN so that data traffic disruption is prevented or minimized when errors occur. The following commands were introduced or modified: **rekey lifetime, rekey retransmit, set security-association lifetime, show crypto gdoi.**



CHAPTER 6

GETVPN Resiliency GM - Error Detection

The GETVPN Resiliency - GM Error Detection feature detects erroneous packets in the data plane for each Group Domain of Interpretation (GDOI) group such as invalid stateful packet inspections (SPIs) or Time-Based Anti-Replay (TBAR) errors. These errors are tracked, and the outer source IP address of the packet is recorded.

Finding Feature Information

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

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

Information About GETVPN Resiliency - GM Error Detection

Error Handling

The GETVPN Resiliency - GM Error Detection feature should be enabled on both the GM and KS for error handling to work. The KS encodes the group information in the SPI (Security Parameter Index) and then it downloads it via the TEK policy to the GM.

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:

```
 *Feb 10 21:01:56.043:
%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
```

The **show crypto gdoi gm** command displays the history of the last 50 Time-Based Anti-Replay (TBAR) errors. You can use these source IP address records to track down the sender group members (GMs) and investigate any existing hardware or software problems. The following statistical information is also available in the command:

- GM recovery feature ON/OFF
- Interval between recoveries
- Number of GM recovery reregistration enforced

When errors occur, the GM reregisters to the next available key server (KS) to retrieve the latest policy and keys and maintains all previously downloaded group policies and keys until the registration is complete.

For instance, when a cooperative key server (COOP KS) split occurs, each promoted KS generates its own Key Encryption Key (KEK) and Traffic Encryption Key (TEK). When a GM receives invalid SPI packets, it will decode it (the KS encodes the group information in the SPI and then it downloads it via the TEK policy to the GM) and if it finds that it belongs to the current getvpn group then it will start the recovery registration.

An invalid SPIs can belong to one of the following two categories:

- Positive invalid SPI: An invalid SPI that belong to the current group and require GM recovery registration.
- Negative invalid SPI: An invalid SPI that does not require recovery registration.

In the case of a positive invalid SPI, a recovery registration to the next key server (KS) on its list is performed. This recovery registration is repeated for each invalid stateful packet inspection (SPI) packet or TBAR error in each client recovery interval to the next KS on the list. When all the KSs in the list are recovered and no longer contain the invalid SPI, that SPI is marked as a false positive and no more recovery registrations are performed. The KSs will always do the recovery registration for TBAR errors. However, once the GM recovers to all the KSs in the list because of an invalid SPI and none of the KSs has that SPI, it will mark that SPI as a false positive and will not do more recovery registrations due to that SPI.

A syslog message is generated to notify you 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.
```

How to Configure GETVPN Resiliency - GM Error Detection

Configuring GETVPN Resiliency - GM Error Detection

SUMMARY STEPS

1. **crypto gdoi group** *group-name*

2. **identity number** *number*
3. **server address ipv4** *address*
4. **client recovery-check interval** *interval*
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GETVPN`	Creates a Group Domain of Interpretation (GDOI) group and enters GDOI group configuration mode.
Step 2	**identity number** *number* **Example:** `Device(config-gdoi-group)# identity number 1111`	Identifies a GDOI group number.
Step 3	**server address ipv4** *address* **Example:** `Device(config-gdoi-group)# server address ipv4 1.0.0.2`	Specifies the IP address of the server that the GDOI group is trying to reach.
Step 4	**client recovery-check interval** *interval* **Example:** `Device(config-gdoi-group)# client recovery-check interval 300`	Sets the interval of time for the client group member (GM) to monitor for control-plane errors.
Step 5	**exit** **Example:** `Device(config-gdoi-group)# exit`	Exits GDOI group configuration mode and returns to global configuration mode.

Configuration Examples for GETVPN Resiliency - GM Error Detection

Example: Configuring GETVPN Resiliency - GM Error Detection

The following example 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
```

Additional References for GETVPN Resiliency - GM Error Detection

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS security commands	*Cisco IOS Security Command References*
Basic deployment guidelines for enabling GET VPN in an enterprise network	*Cisco IOS GET VPN Solutions Deployment Guide*
Designing and implementing a GET VPN network	*Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide*

Standards and RFCs

Standard/RFC	Title
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

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

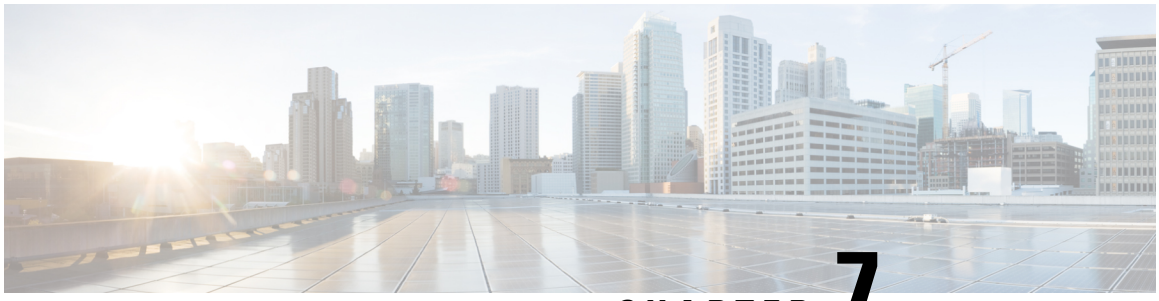
Feature Information for GETVPN Resiliency - GM Error Detection

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

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

Table 10: Feature Information for GETVPN Resiliency - GM Error Detection

Feature Name	Releases	Feature Information
GETVPN Resiliency - GM Error Detection	Cisco IOS XE Release 3.10S	Detects erroneous packets in the data plane for each GDOI group. The following command was introduced: **client recovery-check interval.**



CHAPTER 7

GETVPN CRL Checking

During the Group Encrypted Transport VPN (GET VPN) process, certificates are received from a certificate authority (CA) and used as a proof of identity. Certificates may be revoked for a number of reasons, such as key compromise or certificate loss. Revoked certificates are placed on a certificate revocation list (CRL) that is published periodically to a repository. This list is stored on the repository for the length of time specified by a configured CRL lifetime, and can be anything from a few hours to several days.

Finding Feature Information

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

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

Information About GETVPN CRL Checking

In Internet Key Exchange (IKE), certificates are validated when a session is established between two peers. Current sessions are not affected by certificate revocation. However, new sessions will fail to establish and certificates are not validated again unless group members reregister to the key server (KS).

The GETVPN CRL Checking feature enables public key infrastructure (PKI) to notify Group Domain of Interpretation (GDOI) KSs when a new CRL is available for a configured trustpoint. The KS then 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.

Cooperative Key Server Protocol Integration

Cooperative Key Server Protocol (COOP) is a feature of GET VPN that allows you to configure multiple key servers (KSs) in a VPN network. It is used for KS redundancy.

GETVPN CRL checking integrates with COOP by enabling group member (GM) reauthentication on all KSs. However there is always a possibility that a COOP split may occur, where connectivity is temporarily lost among cooperative KSs.

No COOP Split when Reauthentication is Triggered

If no COOP split occurs the primary GM device deletes the Key Encryption Key (KEK) to secondary KSs and sends a reauthentication message to GMs. The secondary KSs then have the current policies synchronized with the primary policies before the GMs start to reregister. All GMs reregister and reauthenticate to an available KS and receive the new KEK.

COOP Split when Reauthentication is Triggered

If a COOP split occurs before reauthentication is triggered and there are only two primary KSs, they both send out the reauthentication message. Each primary KS creates a new and different KEK. The GM only understands the first reauthentication message it receives as it deletes all the existing KEKs immediately after receiving the message. The GM then reregisters to an available KS and a CRL check takes place. When reregistering, the GM receives either the KEK of the first primary or the KEK of the second primary, depending on which KS the GM reregistered. The GM then installs that KEK and receives further rekeys only from that primary KS. When the COOP merge occurs, the KSs sync up the policies and send rekeys so that all GMs have the current KEK and traffic encryption keys (TEKs).

Avoiding the Creation of Different KEKs

Reauthentication and CRL checking still occurs if reauthentication is triggered during a COOP split. However, triggering the creation of different KEKs in the KSs is avoided by delaying reauthentication. A primary KS only starts the reauthentication if all COOP KSs are reachable (not split). If one COOP KS is not reachable, the primary KS delays sending the reauthentication message until all COOP KSs are reachable.

How to Configure GETVPN CRL Checking

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

- A defined public key infrastructure (PKI) certificate authority (CA) so that group members and key servers are PKI clients and, therefore must enroll to get certificates.
- Key servers (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. See the “Configuring Key Servers for GETVPN CRL Checking” section.
- CRL checking disabled on the group member devices for PKI. See the “Disabling CRL Checking on Group Members” section.
- Internet Key Exchange (IKE) authentication set to certificates. See the “Setting IKE Authentication to Certificates” section

Configuring Key Servers for GETVPN CRL Checking

To configure key servers (KSs) to download the certificate revocation list (CRL) when the first group member (GM) registration occurs after a new CRL is available on the certificate authority (CA), perform the following steps:

SUMMARY STEPS

1. **ip domain name** *name*
2. **ip http server**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **revocation-check** *method*
6. **exit**
7. **crypto identity** *method*
8. **fqdn** *domain*
9. **fqdn** *domain*
10. **exit**
11. **crypto gdoi group** *group-name*
12. **server local**
13. **authorization identity** *name*
14. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ip domain name** *name* **Example:** `Device(config)# ip domain name cisco.com`	Defines a default domain name that the Cisco IOS software uses to complete unqualified hostnames (names without a dotted-decimal domain name).
Step 2	**ip http server** **Example:** `Device(config)# ip http server`	Enables the HTTP server on an IP or IPv6 system.
Step 3	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint mycert`	Defines the trustpoint that your device should use and enters CA trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:** `Device(config-ca-trustpoint)# enrollment url http://10.1.3.1:80`	Specifies the enrollment URL of the CA.

	Command or Action	Purpose
Step 5	**revocation-check** *method* **Example:** `Device(config-ca-trustpoint)# revocation-check crl`	Ensures certificate checking is performed by a CRL.
Step 6	**exit** **Example:** `Device(config-ca-trustpoint)# exit`	Exits CA trustpoint configuration mode and returns to global configuration mode.
Step 7	**crypto identity** *method* **Example:** `Device(config)# crypto identity abcd`	Configures the identity of the device with a given list of distinguished names (DNs) in the certificate of the device and enters crypto identity configuration mode. **Note** You can set restrictions in the device configuration that prevent peers with specific certificates, especially certificates with particular DNs, from having access to selected encrypted interfaces.
Step 8	**fqdn** *domain* **Example:** `Device(config-crypto-identity)# fqdn ut01-unix5.cisco.com`	Derives the name mangler from the remote identity of the fully qualified domain name (FQDN) for a GM.
Step 9	**fqdn** *domain* **Example:** `Device(config-crypto-identity)# fqdn ut01-unix6.cisco.com`	Derives the name mangler from the remote identity of the FQDN for the next GM.
Step 10	**exit** **Example:** `Device(config-crypto-identity)# exit`	Exits crypto identity configuration mode and returns to global configuration mode.
Step 11	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group gdoi-group1`	Creates a Group Domain of Interpretation (GDOI) group and enters GDOI group configuration mode.
Step 12	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.

	Command or Action	Purpose
Step 13	**authorization identity** *name* **Example:** `Device(config-gdoi-local-server)# authorization identity abcd`	Specifies an authorization identity for a GDOI group based on a distinguished name (DN) or FQDN,
Step 14	**end** **Example:** `Device(config-gdoi-local-server)# end`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.

Disabling CRL Checking on Group Members

To disable certificate revocation list (CRL) checking on group members (GMs) for public key infrastructure (PKI), perform the following steps:

SUMMARY STEPS

1. **ip domain name** *name*
2. **ip http server**
3. **crypto pki trustpoint** *name*
4. **enrollment url** *url*
5. **revocation-check** *method*
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**ip domain name** *name* **Example:** `Device(config)# ip domain name cisco.com`	Defines a default domain name that the Cisco IOS software uses to complete unqualified hostnames (names without a dotted-decimal domain name).
Step 2	**ip http server** **Example:** `Device(config)# ip http server`	Enables the HTTP server on an IP or IPv6 system.
Step 3	**crypto pki trustpoint** *name* **Example:** `Device(config)# crypto pki trustpoint mycert`	Defines the trustpoint that your device should use and enters CA trustpoint configuration mode.
Step 4	**enrollment url** *url* **Example:**	Specifies the enrollment URL of the certificate authority (CA).

	Command or Action	Purpose
	`Device(config-ca-trustpoint)# enrollment url http://10.1.3.1:80`	
Step 5	**revocation-check** *method* **Example:** `Device(config-ca-trustpoint)# revocation-check none`	Disables certificate checking on the GMs.
Step 6	**exit** **Example:** `Device(config-ca-trustpoint)# exit`	Exits CA trustpoint mode and returns to global configuration mode.

Setting IKE Authentication to Certificates

SUMMARY STEPS

1. **crypto isakmp policy** *priority*
2. **no authentication pre-share**
3. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto isakmp policy** *priority* **Example:** `Router(config)# crypto isakmp policy 1`	Defines an internet key exchange (IKE) policy and enters ISAKMP policy configuration mode.
Step 2	**no authentication pre-share** **Example:** `Router(config-isakmp)# no authentication pre-share`	Resets the authentication method within the IKE policy to the default value.
Step 3	**end** **Example:** `Router(config)# end`	Returns to privileged EXEC mode.

Enabling GETVPN CRL Checking on Key Servers

To configure public key infrastructure (PKI) to notify the Group Domain of Interpretation (GDOI) key server (KS) when a new certificate revocation list (CRL) is available for the configured trustpoint certificate authority (CA), perform the following steps:

SUMMARY STEPS

1. **crypto gdoi group** *group-name*
2. **server local**
3. **registration periodic crl trustpoint** *trustpoint-name*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group gdoi_group1`	Creates a GDOI group and enters GDOI group configuration mode.
Step 2	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI key server and enters GDOI local server configuration mode.
Step 3	**registration periodic crl trustpoint** *trustpoint-name* **Example:** `Device(config-gdoi-local-server)# registration periodic crl trustpoint mycert`	Enables periodic registrations for the GDOI KSs when new CRLs become available for the configured PKI trustpoint certificate authority.
Step 4	**end** **Example:** `Device(config-gdoi-local-server)# end`	Exits GDOI local server mode and returns to privileged EXEC mode.

Configuration Examples for GETVPN CRL Checking

Example: Enabling GETVPN CRL Checking

The following examples show how the GETVPN CRL checking feature is enabled, including all required preconfigurations.

Example: Configuring Key Servers for GETVPN CRL Checking

In the following example, thekey servers (KSs)are configured to download the certificate revocation list (CRL) when the first group member registration occurs after a new CRL is available on the trustpoint certificate authority (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 gdoi group gdoi-group1
 server local
 authorization identity abcd
```

Example: Disabling CRL Checking on Group Members

In the following example, CRL checking on Group Members (GM) for public key infrastructure (PKI) is disabled:

```
ip domain name cisco.com
ip http server
crypto pki trustpoint mycert
 enrollment url http://10.1.3.1:80
 revocation-check none
```

Example: Setting IKE Authentication to Certificates

```
crypto isakmp policy 1
 no authentication pre-share
```

Example: Enabling GETVPN CRL Checking on Key Servers

In the following example, PKI is configured to notify the GDOI KS named group1 when a new CRL is available for the trustpoint CA named mycert:

```
Crypto gdoi group gdoi_group1
 Server local
  registration periodic crl trustpoint mycert
```

Additional References for GETVPN CRL Checking

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS security commands	*Cisco IOS Security Command References*
Basic deployment guidelines for enabling GET VPN in an enterprise network	*Cisco IOS GETVPN Solution Deployment Guide*
Designing and implementing a GET VPN network	*Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide*

Standards and RFCs

Standard/RFC	Title
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

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

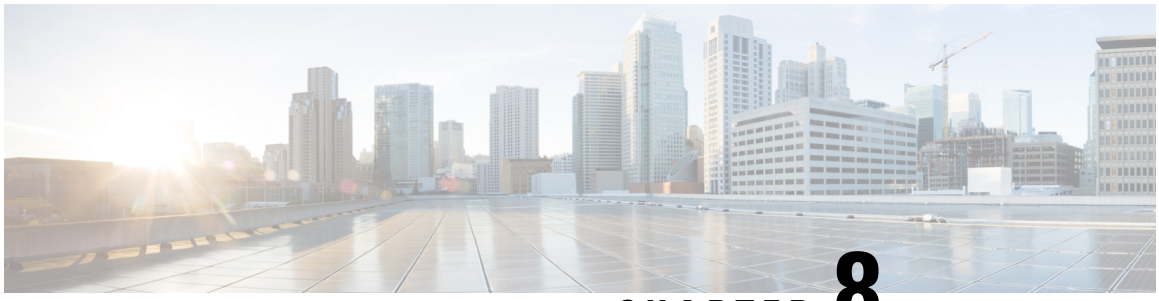
Feature Information for GETVPN CRL Checking

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

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

Table 11: Feature Information for GETVPN CRL Checking

Feature Name	Releases	Feature Information
GETVPN CRL Checking	Cisco IOS XE Release 3.10S	Enables public key infrastructure (PKI) to notify Group Domain of Interpretation (GDOI) key servers (KSs) when a new certificate revocation list (CRL) is available for a configured trustpoint. The following command was introduced: **registration periodic crl trustpoint.**



CHAPTER 8

GET VPN Support with Suite B

The GET VPN Support with Suite B feature adds support of the Suite B set of ciphers to Cisco Group Encrypted Transport (GET) VPN. Suite B is a set of cryptographic algorithms that includes Galois Counter Mode Advanced Encryption Standard (GCM-AES) as well as algorithms for hashing, digital signatures, and key exchange.

Suite B for IP security (IPsec) VPNs is a standard whose usage is defined in RFC 4869, Suite B Cryptographic Suites for IPsec . Suite B provides a comprehensive security enhancement for Cisco IPsec VPNs, and it allows additional security for large-scale deployments. Suite B is the recommended solution for organizations requiring advanced encryption security for the wide-area network (WAN) between remote sites.

Prerequisites for GET VPN Support with Suite B

All key servers (KSs) and group members (GMs) on which you want to enable this feature must be running GET VPN software version 1.0.4 or higher. You should use this feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature. This feature provides a command that you use on the KS (or primary KS) to check whether all devices in the network are running versions that support Suite B. For more information, see the "Ensuring That GMs Are Running Software Versions That Support Suite B" section.

Restrictions for GET VPN Support with Suite B

When they are using a GCM policy or a Galois Message Authentication Code (GMAC) traffic encryption key (TEK) policy, all cooperative KSs for a group must use an access control list (ACL) that has identical ACL entries (ACEs) in the identical order. If not, GMs that register to separate KSs cannot encrypt and decrypt correctly after downloading the policy. This is because with Suite B, an SPI (security parameter index ID that is associated with the TEK) is generated for *each* ACL entry and is unique to each ACL entry.

You cannot reorder entries in an existing ACL. So if you are using a GCM or GMAC TEK policy and must update the ACL on each KS so that it has identical entries in the identical order on each KS, you must remove the ACL from each secondary KS, then create a new ACL on the primary KS, then copy it to the secondary KSs, and then enter the **crypto gdoi ks rekey** command on the primary KS to trigger a rekey across the GET VPN network.

You remove an ACL by using the **no** form of the **ip access-list** command (or if you are using IPv6, the **no** form of the **ipv6 access-list** command).

Suite-B for G-IKEv2 does not work when crypto map is applied on multiple interfaces.

Information About GET VPN Support with Suite B

Suite B

Suite B is standardized by the National Security Agency (NSA) and the National Institute of Standards and Technology (NIST). The GET VPN Support with Suite B feature allows these cryptographic algorithms to be used with GDOI and GET VPN in various ways, including the use of SHA-2/HMAC-SHA-2 and AEC-GCM/AES-GMAC.

Secure Hash Algorithm 2 (SHA-2) is a set of cryptographic hash functions (SHA-224, SHA-256, SHA-384, and SHA-512) designed by the NSA and published by the NIST as a U.S. Federal Information Processing Standard (FIPS). SHA-2 includes many changes from its predecessor, SHA-1. SHA-2 comprises a set of four hash functions with digests that are 224, 256, 384, or 512 bits.

HMAC is a mechanism for message authentication using iterative cryptographic hash functions. HMAC-SHA-2 is HMAC used in combination with the SHA-2 version (SHA-224, SHA-256, SHA-384, and SHA-512) iterative cryptographic hash functions in combination with a secret shared key in IPsec. These combinations are called HMAC-SHA-224, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512. These algorithms can be used as the basis for data origin authentication and integrity verification mechanisms for the authentication header (AH) (although not supported by GET VPN), encapsulating security payload (ESP), IKE, and IKEv2 protocols, and also as pseudo-random functions (PRFs) for IKE and IKEv2.

AES using GCM (AES-GCM) is an encryption algorithm for IPsec. AES using Galois Message Authentication Code (AES-GMAC) is a message integrity algorithm also used for IPsec.

SHA-2 and HMAC-SHA-2

The GET VPN Support with Suite B feature lets you use SHA-2 and HMAC-SHA-2 (HMAC-SHA-256, 384, and 512) as the hash and signature algorithms. SHA-2 and HMAC-SHA-2 with 256, 384, & 512-bit keys are used in

- GDOI registration using IKEv1 as the hash algorithm as described in Section 3.2 (authentication between KSs and GMs) of RFC 6407, The Group Domain of Interpretation .
- The key encryption key (KEK) rekey policy to hash the rekey message for authentication of the rekey message from the KS as well as authentication of the acknowledgment message from the GM.
- The TEK IPsec policy as HMAC-SHA-2 for IPsec SA integrity checking.

AES-GCM and AEC-GMAC

AES-GCM (AES-GCM-128, 192, and 256) and AES-GMAC (AES-GMAC-128, 192, and 256) cryptographic algorithms with 256, 384, and 512-bit keys are used in TEK IPsec policies as IPsec SA encryption and integrity algorithms. GCM is used for encryption and integrity, while GMAC is used for integrity only.

Sets of Cryptographic Algorithms that Comply with Suite B

RFC 4869 describes four sets of cryptographic algorithms for use with IKE and IPsec. When configured, any of these sets will comply with Suite B. Each set consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm:

- Suite-B-GCM-128: Provides ESP integrity protection and confidentiality using 128-bit AES-GCM (see RFC 4106, The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)). Use this suite or Suite-B-GCM-256 when ESP integrity protection and encryption are both needed.
- Suite-B-GCM-256: Provides ESP integrity protection and confidentiality using 256-bit AES-GCM (see RFC 4106, The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)). Use this suite or Suite-B-GCM-128 when ESP integrity protection and encryption are both needed.
- Suite-B-GMAC-128: Provides ESP integrity protection using 128-bit AES-GMAC (see RFC 4543, The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH) but does not provide confidentiality. Use this suite or Suite-B-GMAC-256 only when there is no need for ESP encryption.
- Suite-B-GMAC-256: Provides ESP integrity protection using 256-bit AES-GMAC (see RFC 4543, The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH) but does not provide confidentiality. Use this suite or Suite-B-GMAC-128 only when there is no need for ESP encryption.

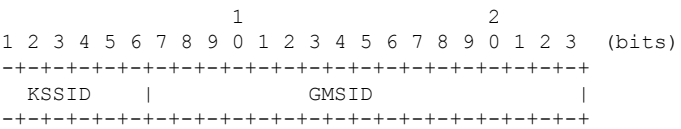
Cisco software contains the ability to configure any of these algorithms. The GET VPN Support with Suite B feature allows GET VPN to use these algorithms.

SID Management

In GET VPN, a counter-based mode of operation (for example, ESP-GCM-AES) requires that an initialization vector (IV) is never reused with a group key. Therefore, this feature provides a method to allow a KS to allocate to each GM (for each interface) a unique sender identifier (SID) for IV construction.

In Suite B, TEK IPsec policies that are used as IPsec SA encryption and integrity algorithms require management of unique pools of SID values on KSs to distribute those unique SID values (GMSIDs) to GMs. Each cooperative KS must have a distinct pool of GMSIDs to allocate. Each KS configures unique KS SIDs (KSSIDs) to configure these SID pools.

A SID space is divided into two parts: a KSSID part and a GMSID part. Therefore, a SID is a concatenation of a KSSID and a GMSID, where the KSSID is the KS portion of a SID, and the GMSID is the GM portion of the SID. A SID is formed by the following bits:



In this example, each KSSID (0 to 127) has 2^{17} (131,072) GMSIDs, which are dynamically assigned to each registering GM.

A GM uses GMSIDs to form a unique 64-bit IV for each packet sent with a given key when using AES-GCM or AES-GMAC. An IV is formed by the following bytes:

```
  0   1   2   3   4   5   6   7   (bytes)
+---+---+---+---+---+---+---+---+
|    SID    |       SSIV       |
+---+---+---+---+---+---+---+---+
```

The sender specific IV (SSIV) is a packet counter.

Group Size

The group size is the length of the SID space allocation for KSSIDs as well as GMSIDs that are reserved to a KS for distribution to GMs. Available group sizes are small (8, 12, or 16 bits), medium (24 bits, which is the default), and large (32 bits). Medium is sufficient for nearly all networks.

You should use a large group size only if you must strictly adhere to the requirement in section A.5, "Key/IV Pair Uniqueness Requirements from SP 800-38D" of the publication Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program in which GET VPN used in conjunction with Suite B must have at least 2^{32} unique possible "module names" (SIDs).This publication is issued and maintained by the NIST and the Communications Security Establishment Canada (CSEC).

For example, in a large group size with one KS, the SID is 32 bits, there are 512 KSSID values (in the range of 0 to 511), and each has 8,388,607 GMSIDs to distribute to registering GMs. With a large group size, use the following KSSID assignment guidelines to configure KSSID ranges:

Table 12: Recommended KSSID Ranges for Group Size Large

KS	1 KS (no cooperative KSs)	2 cooperative KSs	3 cooperative KSs	4 cooperative KSs
KS1	0 - 511	0 - 255	0 - 127	0 - 63
KS2	—	256 - 511	128 - 255	64 - 127
KS3	—	—	256 - 383	128 - 191
KS4	—	—	384 - 511	192 - 255
KS5	—	—	—	256 - 319
KS6	—	—	—	320 - 383
KS7	—	—	—	384 - 447
KS8	—	—	—	448 - 511

If you plan to expand the cooperative KS network to include more KSs, while you are initially configuring the original KS or KSs, use the column in the above table with the *anticipated* number of KSs in the network so that you can add the new KS or KSs later.

You should use a small (8-, 12-, or 16-bit) group size only in well-understood cases where strict interoperability with SID lengths of 8, 12, and 16 bits is required according to RFC 6054, Using Counter Modes with Encapsulating Security Payload (ESP) and Authentication Header (AH) to Protect Group Traffic . If such interoperability is needed, you must be careful when designing the network, because the number of SIDs per

group is severely limited (and therefore, the number of KSs and GMs in a group is severely limited). Following are the limitations for a small group size:

Table 13: Limitations for Group Size Small

SID length	KSSIDs (total KSs)	GMSIDs per KSSID	GMSIDs (total GMs)	Possible number of GM registrations for one KS (after assigning KSSIDs to all KSs evenly)			
—	—	—	—	1 KS	2 KSs	4 KSs	8 KSs
8 bits	2	128	255	320	96	—	—
12 bits	4	1,024	4,095	3,840	1,792	768	—
16 bits	16	4,096	65,535	64,512	31,744	15,360	7,168

KSSID Assignment with Cooperative Key Servers

You should plan ahead to assign a certain number of initial GDOI KS identifiers (KSSIDs) to each KS based on the configured group size, number of KSs, number of GMs, number of GMs per KS, and any future expansion of KSs or GMs (or both).

When there are multiple cooperative KSs in a GDOI group, each KS must have a unique set of KSSID values to ensure that a registering GM never receives the same SID as another registering GM in the group. Therefore, you should plan how you will assign KSSIDs across cooperative KSs in advance, while considering the number of cooperative KSs and if cooperative KSs will be added later. If none will be added, you can assign all available KSSIDs across all KSs. If cooperative KSs will be added, you should reserve some KSSIDs to assign to those KSs when you add them to the network.

You can reassign KSSIDs; however, if KSSIDs that are already used by a KS to distribute GMSIDs are removed from the KS, the group will reinitialize (meaning that all GMs will be forced to re-register, and TEK IPsec SAs will be rekeyed to reset the used KSSIDs) without traffic loss. To avoid this group reinitialization, use the guidelines in the following table (which uses the default group size of medium):

Table 14: Recommended KSSID Assignment Ranges for Cooperative KSs (Group Size Medium)

	1 KS (no cooperative KSs)	2 cooperative KSs	3 cooperative KSs	4 cooperative KSs
KS1	0 - 127	0 - 63	0 - 31	0 - 15
KS2	—	64 - 127	32 - 63	16 - 31
KS3	—	—	64 - 95	32 - 47
KS4	—	—	96 - 127	48 - 64
KS5	—	—	—	65 - 80
KS6	—	—	—	81 - 95
KS7	—	—	—	96 - 112
KS8	—	—	—	113 - 127

If you plan to expand the cooperative KS network to include more KSs, when initially configuring the original KS (or KSs), use the column in the above table with the planned number of KSs in the *expanded* network so that the new KS or KSs can be added later.

Following are additional guidelines for assigning KSSIDs to KSs:

- Configure only contiguous blocks of KSSIDs across KSs (for example, KS1 = 0-9 + 40-49, KS2 = 10-19 + 50-59, KS3 = 20-29, KS4 = 30-39, and so on).
- Any one KS should have enough KSSID space to receive all GM registrations from the group (in case the other KSs fail all of their GM registrations).
- To avoid reinitialization of the group, only add new KSSID values or ranges; do not remove them unless necessary.
- During a network split (a connectivity loss among cooperative KSs), do not change the KSSID assignment; this prevents overlapping KSSIDs, which would cause reinitialization on a merge (when connectivity has been restored among cooperative KSs).
- If the group begins in an *n*-way split (meaning that secondary KSs are planned but not yet configured), configure all of the KSSIDs as if the group was not split.

The number of KSSIDs available depends on the group size configuration as in the following table:

Table 15: Ranges of Available KSSIDs Based on Group Size

Configured Group Size	Number of Available KSSIDs
Small (8-bit)	0 to 1
Small (12-bit)	0 to 3
Small (16-bit)	0 to 15
Medium	0 to 127
Large	0 to 511

Group Reinitialization

Group reinitialization is the process of retiring KSSIDs. Group reinitialization occurs across all KSs (primary and secondary). Any KS can trigger a group reinitialization, and it occurs whenever

- You change the TEK policy from non-GCM to GCM.
- You change the group size.
- You remove a previously used KSSID.
- A KS in the group runs out of both KSSIDs and GMSIDs.
- A KSSID overlap that was detected by a cooperative KS is resolved.

During reinitialization, all KSs move their used KSSIDs to old (used) KSSIDs (and they are thus retired). Then, reinitialization creates a new KEK and new TEKs, lowers the existing TEK lifetime, and deletes the existing TEKs to cause all GMs to re-register (within the window determined by the **clear crypto gdoi ks**

members command). This window is five percent of the remaining lifetime, between 90 seconds and one hour. When the lifetime of the existing TEKs has expired, each KS resets its old (used) KSSIDs, then all KSSIDs are available for use once again.

Reinitialization does not cause traffic disruption on the GMs. All GMs receive new GMSIDs with new TEKs when re-registering.

Cisco GET VPN System Logging Messages for Suite B

The tables below explain the GET VPN system logging (also called syslog) messages that are related to Suite B.

Table 16: KS and Cooperative KS Messages

Message	Explanation
%GDOI-5-KS_REINIT_GROUP: *reason* for group *group-name* and will re-initialize the group.	The KS will reinitialize the group. The possible *reason* strings are as follows: • KS configured Suite-B transform requiring SIDs • KS configured Suite-B transform requiring SIDs during scheduled rekey • KS is running out of SIDs • KS changed Group Size • KS removed used KSSIDs • KS issued 'clear crypto gdoi ks members' • KS issued re-init test cmd • KSSID overlap was resolved • Pri KS peer changed used Group Size • Pri KS peer sent re-init request • Sec KS peer sent re-init request
%GDOI-5-KS_REINIT_FINISH: Re-initialization of group *group-name* completed.	Reinitialization for the group is complete. It is useful to know when a reinitialization has completed, because some operations are blocked during a reinitialization (such as when the group size is changed and used KSSIDs are removed). A reinitialization does not finish until the old (used) TEK is cleared, which might not occur until a reinitialization is checked again (for example while a **show** command is executing, while a group size or KSSIDs are being configured, or when a cooperative KS is being updated) or until the next GM registers.

Message	Explanation
%GDOI-3-KS_NO_SID_AVAILABLE: GMs for group *group-name* need SIDs but this KS has no KS SIDs configured or no more SIDs available.	(When using GCM and after a GM begins registration) GMs for the group need SIDs, but either the KS has no KSSIDs configured or has no more SIDs available.
%GDOI-3-COOP_KS_KSSID_OVERLAP: Overlapping KS Sender Identifier(s) (KSSID) {*KSSID*|*KSSID-Range*} with COOP-KS peer *ip-address* in group *group-name* blocking GM registration (MISCONFIG).	A KSSID or KSSID range that overlaps with a cooperative KS peer in another group is blocking GM registration. An overlapping KSSID configuration is blocked on cooperative KSs by the CLI, but it might occur in a GET VPN network split scenario (in which one or more cooperative KSs were temporarily unavailable but have come back online) or with saved configurations.
%GDOI-5-COOP_KS_KSSID_OVERLAP_RESOLVED: Resolved overlapping KS Sender Identifier(s) (KSSID) with COOP-KS peer allowing GM registrations once again.	A KSSID that overlaps with a cooperative KS peer was resolved (which allows GM registrations to resume).

Table 17: GM Messages

Message	Explanation
%GDOI-5-GM_IV_EXHAUSTED: GM for group *group-name* exhausted its IV space for interface *interface-name* and will re-register.	The GM for the group exhausted its IV space (meaning its set of unique IVs) for a particular SA and will re-register.
%GDOI-5-GM_REJECTING_SA_PAYLOAD: Registration: Policy in SA payload sent by KS *ip-address* rejected by GM in the group *group-name* reason: client rekey hash algorithm (*kek-policy*) is unacceptable by this GM.	The client rekey hash algorithm (the specified KEK policy) was not accepted by a GM in the specified group. At registration, the GM rejected the KEK policy.
%GDOI-5-GM_REJECTING_SA_PAYLOAD: Registration: Policy in SA payload sent by KS *ip-address* rejected by GM in the group *group-name* reason : client rekey transform-sets (*tek-policy*) for data-protection are unacceptable by this GM.	The client rekey transform sets (the specified TEK policy) for data protection was not accepted by the GM. At registration, the GM rejected the TEK policy.
%GDOI-5-GM_REKEY_TRANSFORMSET_CHECK_FAIL: The transform set (*transform-set*) for data protection in group *group-name* is unacceptable by this client.	The transform set for data protection in the group was not accepted by the client. The GM received a rekey and rejected the TEK policy.

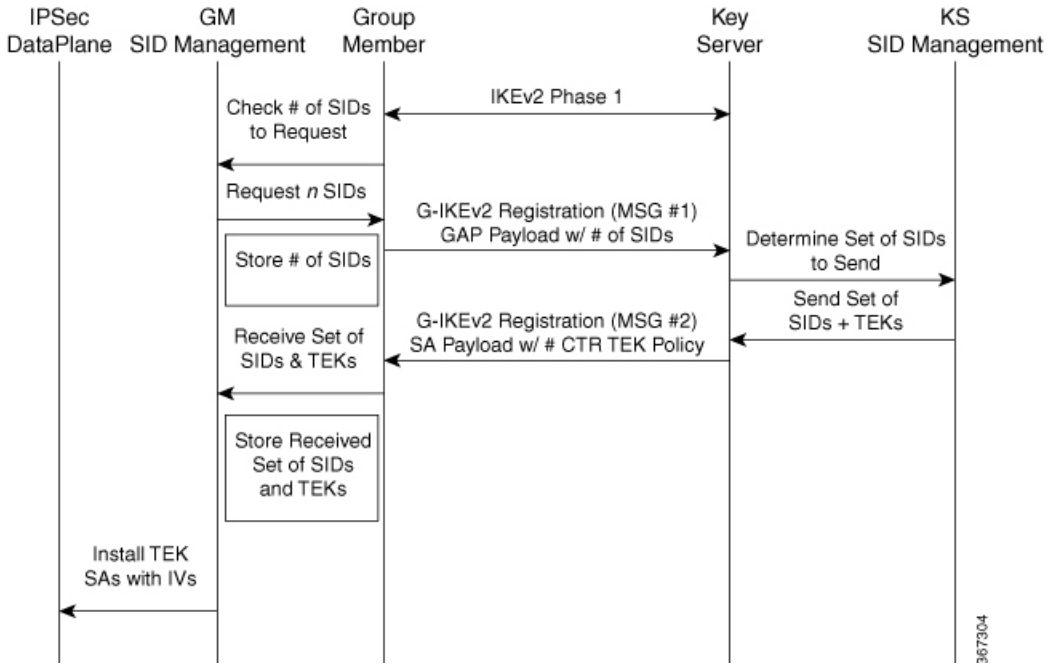
Message	Explanation
%GDOI-3-KS_REKEY_AUTH_KEY_LENGTH_INSUFFICIENT: Rejected rekey sig-hash algorithm change: using sig-hash algorithm HMAC_AUTH_SHA*bits* requires an authentication key length of at least *number-of-bits* bits (*number-of-block*s blocks in bytes) - current RSA key "360-bit" is only 45 blocks in bytes.	Configuration of the rekey signature hash algorithm was rejected, because the RSA key did not have a long enough modulus. HMAC-SHA-384 requires a modulus of at least 465 bits (59 blocks in bytes), and HMAC-SHA-512 requires a modulus of 593 bits (75 blocks in bytes).

Suite B and G-IKEv2

The Ability to use Suite B Algorithms with GIKEv2 with registration interface feature provides support for Suite B on GET VPN G-IKEv2 enabled networks.

The following figure explains the message exchanges between a group member and a key server on GET VPN network enabled with the Ability to use Suite B Algorithms with GIKEv2 with registration interface feature.

Figure 15: Message Exchanges between GM and KS



1. After an IKEv2 session is set up, GM determines the number of SIDs to be requested and sends a registration message to key server via Notify payload requesting the number of SIDs required by group member. At this point, the GM is not aware of the configured lifetime of the TEK SA when requesting SIDs. GM includes the SENDER_ID_REQUEST attribute in the message, irrespective of a CTR transform configuration.
2. KS accepts the Notify payload containing SENDER_ID_REQUEST attribute in the registration message and sends SA payload to GM.

3. If the **crypto ipsec transform-set** command is configured, key server sends KD payload containing the number of requested SIDs. If the **crypto ipsec transform-set** command is not configured, key server will not send SIDs, even though GM requests for SIDs.

Note If no SIDs are requested and if the **crypto ipsec transform-set** command is configured, the KS will send one SID, which is the default SID value.

Working of a Group Member with Suite B and G-IKEv2

GM must receive and install SIDs after a successful registration. To support Suite-B with G-IKEv2, a GM must do the following:

- Send Notify payload during GM registration requesting a number of SIDs.
- Determine the number of SIDs required based on the number of client registration interfaces to which crypto map is applied on GM.
- Receive KD payload from KS. KD payload contains SIDs sent from KS.
- Install TEK SAs with one or more SIDs. The number of SIDs + initial SSIV = IV.
- Reregister when SSIV exhausts, when no SIDs exist.

IPsec installs the SAs after receiving TEK SA with IV values in the KMI message from G-IKEv2 to IPsec.

Working of a Key Server with Suite B and G-IKEv2

To support Suite-B with G-IKEv2, KS must do the following:

- Receive and process Notify payload from GM during registration.
- If Notify payload is not received and the **crypto ipsec transform-set** command is configured, assign one unique SID to each GM.
- After receiving the SID request, the number of SIDs sent to GM is calculated based on the configured TEK SA lifetime.
- If SA lifetime is less than or equal to 1 day (86400 seconds), the number of SIDs that KS sends to GM will be calculated as follows:

  Number of SIDs KS sends = Number of SIDs GM requested

- If SA lifetime is more than 1 day, the number of SIDs that KS sends to GM will be calculated as follows. SIDs are specific to SA lifetime:

  Number of SIDs = Number of SIDs requested by GM * ceil (configured SA lifetime in KS/ 86400)

- SIDs are divided among the available crypto map interfaces and sent to GM via KD payload.

The following is a sample output of the **show gdoi gm identifier detail** command that displays the SID distribution if SA lifetime is greater than 1 day:

```
Device# show cry gdoi gm identifier detail

GM Sender ID (SID) Information for Group GKM-GROUP-KS_KDN:

  Group Member: 10.10.10.2       vrf: None
    Transform Mode                  : Counter (Suite-B)
    # of SIDs Last Requested        : 3

    CURRENT SIDs:
      Shared Across Interfaces?     : No
      SID Length (Group Size)       : 24 bits (MEDIUM)
      # of SIDs Downloaded          : 6
      First SID Downloaded          : 0x00000001
      Last SID Downloaded           : 0x00000006

      CM Interface       Packets / Sec   # Req # Rx   Installed SID Range
      ================== =============== ===== ====== ======================
      Ethernet0/0        16842           1     2      0x00000001 - 0x00000002
      Ethernet0/1        16842           1     2      0x00000003 - 0x00000004
      Ethernet0/2        16842           1     2      0x00000005 - 0x00000006

    NEXT SID REQUEST:
      TEK Lifetime                  : 86453 sec
      SID Length (Group Size)       : 24 bits (MEDIUM)
```

How to Configure GET VPN Support with Suite B

Each feature in the GET VPN Support with Suite B feature set is independently configurable. But to be compliant with the Suite B standard, you must configure certain combinations of these features. For more information about these combinations, see RFC 4869, Suite B Cryptographic Suites for IPsec .

Ensuring that GMs Are Running Software Versions That Support Suite B

Because GET VPN is a technology that is based on groups, all devices in the same group (including the primary KS, cooperative KSs, and GMs) must support the Suite B feature before you can enable the feature. If you want to enable the feature for a group, you must ensure that all devices in the group are running compatible versions of the GET VPN software.

To ensure that all devices in the GET VPN network support Suite B, perform the following steps on the KS (or primary KS).

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature suite-b**
3. **show crypto gdoi feature suite-b | include No**

DETAILED STEPS

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

	Command or Action	Purpose
	`Device> enable`	
Step 2	**show crypto gdoi feature suite-b** **Example:** `Device# show crypto gdoi feature suite-b`	Displays the version of the GET VPN software running on each KS and GM in the network and displays whether that device supports Suite B.
Step 3	**show crypto gdoi feature suite-b | include No** **Example:** `Device# show crypto gdoi feature suite-b | include No`	(Optional) Finds only those devices that do not support Suite B.

Configuring a Key Server for GET VPN Suite B

Configuring the Signature Hash Algorithm for the KEK

Perform this task to configure the signature hash algorithm for the KEK.

Before you begin

This task has the following prerequisites:

- Make sure that rekey authentication that is using an RSA key pair associated with the device is enabled. To do so, use the **rekey authentication** command with the **mypubkey rsa** *key-name* keywords and argument.
- Make sure that the RSA key pair has a modulus of sufficient length. HMAC-SHA-384 requires a modulus of at least 465 bits (59 blocks in bytes), and HMAC-SHA-512 requires a modulus of 593 bits (75 blocks in bytes). If the rekey signature hash algorithm is changed to SHA-384 or SHA-512 with a key pair of insufficient modulus length, a configuration rejection message appears on the console, and system logging messages are generated. Similarly, if the rekey signature hash algorithm is already SHA-384 or SHA-512 and the key pair is modified to one of insufficient modulus length, a similar message appears on the console, and the same system logging messages are generated.
- To use SHA-2/HMAC-SHA-2 for authentication of the *acknowledgment* from GMs to KSs after receiving a rekey message, you must enable unicast distribution of rekey messages to GMs. To do so, use the **rekey transport unicast** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** [**ipv6**] *group-name*
4. **server local**
5. **rekey sig-hash algorithm** *algorithm*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group mygroup`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.
Step 4	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI KS and enters GDOI local server configuration mode.
Step 5	**rekey sig-hash algorithm** *algorithm* **Example:** `Device(gdoi-local-server)# rekey sig-hash algorithm sha512`	Configures the signature hash algorithm for the KEK. For Suite B, you must specify **sha256**, **sha384**, or **sha512**.
Step 6	**end** **Example:** `Device(gdoi-local-server)# end`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.

Configuring the Group Size

This task is optional. For nearly all deployments, the default group size (sender identifier length) of medium is recommended. Perform this task to configure the group size for Suite B.

When you change the group size in a group with cooperative KSs after Suite B (meaning ESP-GCM or ESP-GMAC) is configured and after the Suite B policy has been generated, you must change the group size on all secondary KSs before changing it on the primary KS.

Changing the group size causes the group to reinitialize (so that the new SID length can be used). Conflicting group size configurations across KSs will block GM registration.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** [**ipv6**] *group-name*
4. **server local**
5. **group size** {**small** {**8** | **12** | **16**} | **medium** | **large**}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group mygroup`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.
Step 4	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI KS and enters GDOI local server configuration mode.
Step 5	**group size** {**small** {**8**	**12**	**16**}	**medium**	**large**} **Example:** `Device(gdoi-local-server)# group size small 16`	Configures the group size.
Step 6	**end** **Example:** `Device(gdoi-local-server)# end`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.

Configuring Key Server Identifiers

Suite B requires the assignment of unique GMSIDs to each GM, which means that a GM cannot reuse a previously used SID (either from itself or another GM) for the same key. Therefore, although GET VPN is

designed to disallow overlapping SID values, you should correctly configure KSSID values among KSs so that each KS has a unique set. (KSSID overlap among KSs will cause a reinitialization.)

You must configure at least one unique KSSID to allot a pool of SIDs to the KS. You do so on the KS before configuring GCM or GMAC as the TEK IPsec policy.

Perform this task to assign a KSSID or a range of KSSIDs to a KS. Each KS must be assigned at least one KSSID when using GCM or GMAC. You can configure a single KSSID, a range of KSSIDs, or both. For the default group size of medium, there are 128 possible KSSID values in the range from 0 to 127.

KSSID values are not assigned to (and usable by) the KS until you exit GDOI local server ID configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** [**ipv6**] *group-name*
4. **server local**
5. **identifier**
6. **range** *lowest-kssid* **-** *highest-kssid*
7. **value** *kssid*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group mygroup`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.
Step 4	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI KS and enters GDOI local server configuration mode.
Step 5	**identifier** **Example:**	Enters GDOI local server ID configuration mode.

	Command or Action	Purpose
	`Device(gdoi-local-server)# identifier`	
Step 6	**range** *lowest-kssid - highest-kssid* **Example:** `Device(gdoi-local-server-id)# range 10 - 20`	Assigns a range of KSSIDs. • This range must be unique in the entire group.
Step 7	**value** *kssid* **Example:** `Device(gdoi-local-server-id)# value 0`	Assigns a KSSID. • This KSSID must be unique in the entire group. • The **value 0** command allots the pool of SIDs to the KS that begin with KSSID value 0 (meaning that it is allotted the pool of SID values beginning with 0x0 and ending with 0x1FFFF).
Step 8	**end** **Example:** `Device(gdoi-local-server-id)# end`	Exits GDOI local server ID configuration mode and returns to privileged EXEC mode.

If you try to configure one or more KSSIDs on a KS that are already assigned to another KS (and the cooperative KS network is not split), the configuration is denied, and the following message appears when you exit GDOI local server ID configuration mode:

```
% Key Server SID Configuration Denied:
%   The following Key Server SIDs being added overlap:
%   2, 200-250  (COOP-KS Peer: 10.0.9.1)
```

If the cooperative KS network *is* split, you should not configure overlapping KSSIDs. If overlapping KSSIDs are detected on a network merge, GM registration is blocked until the overlap is resolved. The following system logging message appears on both KSs:

```
%GDOI-3-COOP_KSSID_OVERLAP: Overlapping KS Sender Identifier(s) (KSSID) {2, 200-250} with
COOP-KS peer 10.0.9.1 in group diffint blocking GM registration (MISCONFIG)
```

When a KS unconfigures the overlapping KSSIDs, the group reinitializes (meaning that all GMs are forced to re-register, and TEK IPsec SAs are rekeyed to reset the used KSSIDs) without traffic loss. The following system logging messages appear on the KS:

```
%SYS-5-CONFIG_I: Configured from console by console
%GDOI-5-COOP_KSSID_OVERLAP_RESOLVED: Resolved overlapping KS Sender Identifier(s) (KSSID)
with COOP-KS peer allowing GM registrations once again
%GDOI-5-KS_REINIT_GROUP: KSSID overlap was resolved for group diffint and will re-initialize
 the group.
%GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey for group diffint from address 10.0.8.1
 with seq # 11
%GDOI-4-GM_DELETE: GM 10.0.3.1 deleted from group diffint.
%GDOI-4-GM_DELETE: GM 10.65.9.2 deleted from group diffint.
```

The %GDOI-5-KS_SEND_UNICAST_REKEY system logging message appears only if this is the primary KS. The peer KS that had overlapping KSSIDs also displays the %GDOI-5-COOP_KSSID_OVERLAP_RESOLVED system logging message.

Configuring the IPsec SA for Suite B

To configure the IPsec SA for Suite B, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name* {**esp-gcm** | **esp-gmac**} [**128** | **192** | **256**]
4. **crypto ipsec profile** *ipsec-profile-name*
5. **set transform-set** *transform-set-name*
6. **exit**
7. **crypto gdoi group** [**ipv6**] *group-name*
8. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
9. **server local**
10. **sa ipsec** *sequence-number*
11. **profile** *ipsec-profile-name*
12. **match address** {**ipv4** | **ipv6**} {*access-list-number* | *access-list-name*}
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name* {**esp-gcm**	**esp-gmac**} [**128**	**192**	**256**] **Example:** `Device(config)# crypto ipsec transform-set g1 esp-gcm 192`	Defines a transform set—an acceptable combination of security protocols and algorithms—and enters crypto transform configuration mode. • For Suite B, you must specify a transform set using ESP-GCM or ESP-GMAC. (You can define multiple transform sets by entering the command again on separate command lines.)

	Command or Action	Purpose
		• You can optionally specify a key size of 128, 192, or 256. The default key size is 128.
Step 4	**crypto ipsec profile** *ipsec-profile-name* **Example:** `Device(config)# crypto ipsec profile profile1`	Defines the IPsec profile (the parameters to be used for IPsec encryption between two IPsec routers) and enters IPsec profile configuration mode.
Step 5	**set transform-set** *transform-set-name* **Example:** `Device(ipsec-profile)# set transform-set transformset1`	Specifies which transform sets can be used with the crypto map entry.
Step 6	**exit** **Example:** `Device(ipsec-profile)# exit`	Exits IPsec profile configuration mode.
Step 7	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group gdoigroupname`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.

Step 8	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Device(config-gdoi-group)# identity number 3333`
	Example:
	`Device(config-gdoi-group)# identity address ipv4 209.165.200.225`
		Identifies a GDOI group number or address.
		• The **identity number** *number* command applies to IPv4 and IPv6 configurations.
		• The **identity address ipv4** *address* command applies only to IPv4 configurations.

Step 9	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI KS and enters GDOI local server configuration mode.
Step 10	**sa ipsec** *sequence-number* **Example:** `Device(gdoi-local-server)# sa ipsec 1`	Specifies the IPsec SA policy information to be used for a GDOI group and enters GDOI SA IPsec configuration mode.
Step 11	**profile** *ipsec-profile-name* **Example:**	Defines the IPsec SA policy for a GDOI group.

	Command or Action	Purpose
	`Device(gdoi-sa-ipsec)# profile gdoi-p`	

Step 12	**match address** {**ipv4** | **ipv6**} {*access-list-number* | *access-list-name*}
	Example:
	`Device(gdoi-sa-ipsec)# match address ipv4 102`
		Selects an IP extended access list (ACL) for a GDOI registration.
		• You must use the **ipv4** keyword for IPv4 groups and the **ipv6** keyword for IPv6 groups.
		• You must use a named (not numbered) access list for IPv6 configurations.
		Note Make sure that you select an ACL that has identical entries in the identical order among all the cooperative KSs for the group. If not, GMs that register to separate KSs cannot encrypt and decrypt correctly after downloading the policy.
		Note If you attempt to assign an IPv6 group with IPv4 policies, an error message appears indicating that the access list name is invalid, or the list already exists but is the wrong type:
		`Access-list type conflicts with prior definition`
		`% ERROR: access-list-name is either an invalid name or`
		`the list already exists but is the wrong type.`

Step 13	**end** **Example:** `Device(gdoi-sa-ipsec)# end`	Exits GDOI SA IPsec configuration mode and returns to privileged EXEC mode.

Configuring a Group Member for GET VPN Suite B

Configuring Acceptable Ciphers or Hash Algorithms for KEK for Suite B

To configure the Suite B ciphers and hash algorithms for KEK to be allowed by the GM, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** [**ipv6**] *group-name*
4. Enter one of the following commands:

- **identity number** *number*
- **identity address ipv4** *address*

5. **server address ipv4** *address*
6. **client rekey encryption** *cipher* [... [*cipher*]]
7. **client rekey hash** *hash*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Device(config-gdoi-group)# identity number 3333`
	Example:
	`Device(config-gdoi-group)# identity address ipv4 10.2.2.2`
		Identifies a GDOI group number or address.

Step 5	**server address ipv4** *address* **Example:** `Device(config-gdoi-group)# server address ipv4 10.0.5.2`	Specifies the address of the server that a GDOI group is trying to reach. • To disable the address, use the **no** form of the command.
Step 6	**client rekey encryption** *cipher* [... [*cipher*]] **Example:** `Device(config-gdoi-group)# client rekey encryption 3des-cbc aes 192 aes 256`	Sets the client acceptable rekey ciphers for the KEK.

	Command or Action	Purpose
Step 7	**client rekey hash** *hash* **Example:** `Device(config-gdoi-group)# client rekey hash sha384`	Sets the client acceptable hash algorithm for KEK. • For Suite B, you must specify either **sha256**, **sha384**, or **sha512**.
Step 8	**end** **Example:** `Device(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Configuring Acceptable Transform Sets for TEKs for Suite B

To configure the transform sets used by TEKs for data encryption or authentication to be allowed by the GM, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ipsec transform-set** *transform-set-name* {**esp-gcm** | **esp-gmac**} [**128** | **192** | **256**]
4. **exit**
5. **crypto gdoi group** [**ipv6**] *group-name*
6. **client transform-sets** *transform-set-name1* [... [*transform-set-name6*]]
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ipsec transform-set** *transform-set-name* {**esp-gcm**	**esp-gmac**} [**128**	**192**	**256**] **Example:** `Device(config)# crypto ipsec transform-set g1 esp-gcm 192`	Defines a transform set—an acceptable combination of security protocols and algorithms—and enters crypto transform configuration mode. • For Suite B, you must specify a transform set using ESP-GCM or ESP-GMAC. • You can define multiple transform sets by entering the command again on separate command lines.

	Command or Action	Purpose
		• You can optionally specify a key size of 128, 192, or 256. The default key size is 128.
Step 4	**exit** **Example:** `Device(cfg-crypto-trans)# exit`	Exits crypto transform configuration mode.
Step 5	**crypto gdoi group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gdoi group gdoigroupone`	Identifies a GDOI group and enters GDOI group configuration mode. • If you are using GET VPN over IPv6 in the data plane, you must use the **ipv6** keyword.
Step 6	**client transform-sets** *transform-set-name1* [... [*transform-set-name6*]] **Example:** `Device(config-gdoi-group)# client transform-sets g1`	Specifies the acceptable transform-set tags used by TEKs for data encryption and authentication. • You can specify up to six transform-set tags.
Step 7	**end** **Example:** `Device(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Verifying and Troubleshooting GET VPN Support with Suite B

Verifying and Troubleshooting GET VPN Support with Suite B on a Key Server

To view the configuration that is running on a KS, use the **show running-config** command.

SUMMARY STEPS

1. **show crypto gdoi ks identifier** [**detail**]
2. **show crypto gdoi ks coop identifier** [**detail**]
3. **show crypto gdoi feature suite-b**
4. **show crypto gdoi ks policy**

DETAILED STEPS

Step 1 **show crypto gdoi ks identifier** [**detail**]

Example:

```
Device# show crypto gdoi ks identifier detail
```

```
KS Sender ID (KSSID) Information for Group diffint:

    Transform Mode           : Counter (Suite B)
    reinitializing          : No
    SID Length (Group Size)  : 24 bits (medium)
    Current KSSID In-Use     : 0
    Last GMSID Used          : 1

    KSSID (or SIDS)Assigned        : 0-15
    KSSID (or SIDS)Used            : 0
    KSSID (or SIDS) Used  (Old)      : none
    Available KSSID (or SIDS): 1-15

    REMAINING SIDs:
      KSSID to reinitialize at         : 15
      GMSID to reinitialize at         : 6291456
      # of SIDs Remaining for Cur KSSID : 8388606
      # of SIDs Remaining until Re-init : 132120575
```

This command displays the status of SID management for Suite B. The Transform Mode field can be either Non-Counter (Non-Suite B) or Counter (Suite B) to check if SID management and a Suite B policy is currently used in the group. If the group is currently reinitializing (meaning that all GMs will be forced to re-register, and TEK IPsec SAs will be rekeyed to reset the used KSSIDs), then the reinitializing field displays Yes. The SID Length (Group Size) field determines the group size currently used in the group, which defaults to 24 bits (medium).

The Current KSSID In-Use and Last GMSID Used fields correspond to the SID (or SIDS) to be distributed to the next registering GM. The KSSID (or SIDS) Assigned field corresponds to the locally configured KSSIDs that have been synced with cooperative KSs, and the Available KSSID (or SIDS) field corresponds to those KSSIDs that have not been used yet since the last reinitialization. Each time a new KSSID is used, it is added to the KSSID (or SIDS) Used field, and during a reinitialization, those used KSSIDs are transferred to the KSSID (or SIDS) Used (Old) field. At the end of a reinitialization period, the old used KSSIDs are cleared and put in the Available KSSIDs pool again.

Note When the value in the # of SIDs Remaining until Re-init field approaches 0, a reinitialization will occur soon if GMs are continuing to re-register. Although a reinitialization should not cause traffic disruption or network problems, it will cause all GMs to re-register.

Step 2 **show crypto gdoi ks coop identifier** [**detail**]

Example:

```
Device# show crypto gdoi ks coop identifier detail

COOP-KS Sender ID (SID) Information for Group diffint:

    Local KS Role: Primary   , Local KS Status: Alive
        Local Address                : 10.0.8.1
        Next SID Client Operation    : NOTIFY
        reinitializing              : No
        KSSID Overlap                : No
        SID Length (Group Size) Cfg  : 24 bits (medium)
        SID Length (Group Size) Used : 24 bits (medium)
        Current KSSID In-Use         : 0
        KSSID (or SIDS)Assigned            : 0-15
        KSSID (or SIDS)Used                : 0
        Old KSSID (or SIDS)Used            : none

    Peer KS Role:  Secondary , Peer KS Status:  Alive
        Peer Address                 : 10.0.9.1
        Next SID Client Operation    : NOTIFY
```

```
reinitializing              : No
KSSID Overlap                : No
SID Length (Group Size) Cfg  : 24 bits (medium)
SID Length (Group Size) Used : 24 bits (medium)
Current KSSID In-Use         : 16
KSSID (or SIDS)Assigned           : 16-31
KSSID (or SIDS)Used               : 16
Old KSSID (or SIDS)Used           : none
```

This command displays the status of SID information that is synchronized across cooperative KSs.

When the KSSID Overlap field displays Yes, GM registration is blocked until the overlap of KSSIDs (which could have happened during a network split) is resolved. You must unconfigure the overlapping KSSIDs from one cooperative KS or the other before GM registration can resume. When the overlapping KSSIDs are resolved, a reinitialization occurs.

When you change the group size (not recommended for most deployments), all secondary KSs must first configure the new group size. Then on the primary KS, the SID Length (Group Size) Cfg field displays the new group size on all cooperative KS peers. Only when the primary KS configures the new group size will all KSs start to use the new group size and update the SID Length (Group Size) Used field to display the new group size.

Step 3 **show crypto gdoi feature suite-b**

Example:

```
Device# show crypto gdoi feature suite-b

Group Name: diffint
    Key Server ID       Version   Feature Supported
        10.0.8.1         1.0.4          Yes
        10.0.9.1         1.0.4          Yes

    Group Member ID     Version   Feature Supported
        10.0.3.1         1.0.4          Yes
        10.0.4.1         1.0.4          Yes
```

This command displays whether KSs and GMs can use the Suite B feature set (meaning AES-GCM, AES-GMAC, SHA-2, and HMAC-SHA2). The Version field must display 1.0.4 or higher, and the Feature Supported field must display Yes for all KSs in the cooperative KS group and for the registered GMs.

Step 4 **show crypto gdoi ks policy**

Example:

```
Device# show crypto gdoi ks policy

Key Server Policy:
For group diffint (handle: 2147483650) server 10.0.8.1 (handle: 2147483650):

  # of teks : 4  Seq num : 0
  KEK POLICY (transport type : Unicast)
    spi : 0x80474E999FE8F60364B7F51809E28C84
    management alg     : disabled    encrypt alg       : 3DES
    crypto iv length   : 8           key size          : 24
    orig life(sec): 86400       remaining life(sec): 85586
    sig hash algorithm : enabled     sig key length    : 162
    sig size           : 128
    sig key name       : mykeys
```

```
TEK POLICY (encaps : ENCAPS_TUNNEL)
  spi               : 0x9C666FA7
  access-list       : gcm-acl
  Selector          : permit ip host 10.0.1.1 host 239.0.1.1
  transform         : esp-gcm
  alg key size      : 20            sig key size          : 0
  orig life(sec)    : 900           remaining life(sec)   : 87
  tek life(sec)     : 900           elapsed time(sec)     : 813
  override life (sec): 0            antireplay window size: 64


TEK POLICY (encaps : ENCAPS_TUNNEL)
  spi               : 0x54E8D5D3
  access-list       : gcm-acl
  Selector          : permit ip host 10.0.100.2 host 238.0.1.1
  transform         : esp-gcm
  alg key size      : 20            sig key size          : 0
  orig life(sec)    : 900           remaining life(sec)   : 87
  tek life(sec)     : 900           elapsed time(sec)     : 813
  override life (sec): 0            antireplay window size: 64



TEK POLICY (encaps : ENCAPS_TUNNEL)
  spi               : 0xC8B4DE6D
  access-list       : gcm-acl
  Selector          : permit ip host 10.0.1.1 host 10.0.100.2
  transform         : esp-gcm
  alg key size      : 20            sig key size          : 0
  orig life(sec)    : 900           remaining life(sec)   : 87
  tek life(sec)     : 900           elapsed time(sec)     : 813
  override life (sec): 0            antireplay window size: 64


TEK POLICY (encaps : ENCAPS_TUNNEL)
  spi               : 0x1C908AF3
  access-list       : gcm-acl
  Selector          : permit ip host 10.0.100.2 host 10.0.1.1
  transform         : esp-gcm
  alg key size      : 20            sig key size          : 0
  orig life(sec)    : 900           remaining life(sec)   : 87
  tek life(sec)     : 900           elapsed time(sec)     : 813
```

This command displays whether a TEK and IPsec SA were generated per ACE (displayed in the Selector field) from the ACL in the access-list field for the ESP-GCM or ESP-GMAC TEK policy. This command also displays whether the KEK policy is using SHA-2/HMAC-SHA-2 as the signature hash algorithm.

Verifying and Troubleshooting GET VPN Support with Suite B on a GM

To view the configuration that is running on a GM, use the **show running-config** command.

SUMMARY STEPS

1. **show crypto gdoi gm identifier** [**detail**]
2. **show crypto gdoi feature suite-b**
3. **show crypto gdoi**

DETAILED STEPS

Step 1 **show crypto gdoi gm identifier** [**detail**]

Example:

```
Device# show crypto gdoi gm identifier detail

GM Sender ID (SID) Information for Group diffint:

  Group Member: 10.65.9.2        vrf: None
    Transform Mode                  : Counter (Suite B)
    # of SIDs Last Requested        : 3

    CURRENT SIDs:
      Shared Across Interfaces?     : Yes
      SID Length (Group Size)       : 24 bits (medium)
      # of SIDs Downloaded          : 3
      First SID Downloaded          : 0x08000007
      Last SID Downloaded           : 0x08000009

      CM Interface   B/W (Kbps) MTU (B)  # Req # Rx  Installed SID Range
      ============== ========== ======== ===== ===== ======================
      Et2/0          10000      1500     1     3     0x08000007 - 0x08000009
      Et3/0          10000      1500     1     3     0x08000007 - 0x08000009
      Et4/0          10000      1500     1     3     0x08000007 - 0x08000009

    NEXT SID REQUEST:
      TEK Lifetime                  : 900 sec
      SID Length (Group Size)       : 32 bits (LARGE)
```

This command displays the status of received and installed SIDs on a GM when it is using GCM-AES or GMAC-AES as the TEK IPsec SA policy. The Transform Mode field can display Non-Counter (Non-Suite B) or Counter (Suite B) to check whether SIDs are being downloaded and installed and whether a Suite B policy is used in the group. The # of SIDs Last Requested field mainly depends on the number of interfaces to which the crypto map is applied for this registered GM (meaning using the local-address or client registration interface). The SIDs are Shared Across Interfaces field when using local-address and each CM Interface's Installed SID Range field will be the same. You use this command mainly to verify that each CM interface has SIDs installed.

Step 2 **show crypto gdoi feature suite-b**

Example:

```
Device# show crypto gdoi feature Suite B

       Version    Feature Supported
        1.0.4           Yes
```

This command displays whether this GM can use the Suite B feature set (meaning GCM-AES, GMAC-AES, SHA-2, and HMAC-SHA-2). The Version field must display 1.0.4 or higher, and the Feature Supported field must display Yes.

Step 3 **show crypto gdoi**

Example:

```
Device# show crypto gdoi

GROUP INFORMATION

    Group Name               : diffint
    Group Identity           : 1234
    Crypto Path              : ipv4
    Key Management Path      : ipv4
    Rekeys received          : 0
    IPSec SA Direction       : Both

     Group Server list       : 10.0.8.1

    Group member             : 10.0.3.1         vrf: None
       Version               : 1.0.4
       Registration status   : Registered
       Registered with       : 10.0.8.1
.
.
.
ACL Downloaded From KS 10.0.8.1:
   access-list   permit ip host 10.0.1.1 host 239.0.1.1
   access-list   permit ip host 10.0.100.2 host 238.0.1.1
   access-list   permit ip host 10.0.1.1 host 10.0.100.2
   access-list   permit ip host 10.0.100.2 host 10.0.1.1

KEK POLICY:
    Rekey Transport Type     : Unicast
    Lifetime (secs)          : 85740
    Encrypt Algorithm        : 3DES
    Key Size                 : 192
    Sig Hash Algorithm       : HMAC_AUTH_SHA256
    Sig Key Length (bits)    : 1024

TEK POLICY for the current KS-Policy ACEs Downloaded:
  Ethernet3/0:
    IPsec SA:
        spi: 0x318846DE(831014622)
        transform: esp-gcm
        sa timing:remaining key lifetime (sec): (86350)
        Anti-Replay(Counter Based) : 64

    IPsec SA:
        spi: 0xF367AEA0(4083658400)
        transform: esp-gcm
        sa timing:remaining key lifetime (sec): (86350)
        Anti-Replay(Counter Based) : 64

    IPsec SA:
        spi: 0xE583A3F5(3850609653)
        transform: esp-gcm
        sa timing:remaining key lifetime (sec): (86350)
        Anti-Replay(Counter Based) : 64

    IPsec SA:
        spi: 0xE9AC04C(245022796)
        transform: esp-gcm
        sa timing:remaining key lifetime (sec): (86350)
        Anti-Replay(Counter Based) : 64
```

The presence of multiple IPsec SAs shows that GCM or GMAC is configured (note that each IPsec SA has a unique SPI for each ACE that was downloaded). For each ACE listed in the TEK POLICY for the current KS-Policy ACEs Downloaded

section, this command displays whether a TEK policy and IPsec SA were downloaded (and installed) from the ACLs that are listed in the ACL Downloaded From KS section. This command also displays whether the KEK policy is using SHA-2/HMAC-SHA-2 for the signature hash algorithm (for example, HMAC_AUTH_SHA256).

Configuration Examples for GET VPN Support with Suite B

Example: Ensuring that GMs Are Running Software Versions That Support Suite B

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to check whether all the devices in each group support Suite B cryptography:

```
Device# show crypto gdoi feature suite-b

Group Name: GETVPN
    Key Server ID       Version   Feature Supported
        10.0.5.2         1.0.4          Yes
        10.0.6.2         1.0.4          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         1.0.4          Yes
        10.0.3.2         1.0.4          Yes
```

You can also enter the above command on a GM (which will display 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 GET VPN network that do *not* support Suite B:

```
Device# show crypto gdoi feature suite-b | 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
```

Example: Configuring a Key Server for GET VPN Suite B

Configuring the Signature Hash Algorithm for the KEK

The following example shows how to configure the signature hash algorithm for the KEK:

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group mygroup
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# rekey sig-hash algorithm sha512
Device(gdoi-local-server)# end
```

Configuring the Group Size for Suite B

Configuring the group size for Suite B is optional, because the default group size of medium is sufficient for most deployments. The following example shows how to configure the group size for Suite B:

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group mygroup
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# group size small 16
Device(gdoi-local-server)# end
```

Configuring Key Server Identifiers

The following example shows how to assign a KSSID as well as a range of KSSIDs to a KS:

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group mygroup
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# identifier
Device(gdoi-local-server-id)# range 10 - 20
Device(gdoi-local-server-id)# value 0
Device(gdoi-local-server-id)# end
```

Configuring the IPsec SA for Suite B

The following example shows how to configure the IPsec SA for Suite B. This example uses an identity number instead of an identity address:

```
Device> enable
Device# configure terminal
Device(config)# crypto ipsec transform-set g1 esp-gcm 192
Device(config)# crypto ipsec profile profile1
Device(ipsec-profile)# set transform-set transformset1
Device(ipsec-profile)# exit
Device(config)# crypto gdoi group gdoigroupname
Device(config-gdoi-group)# identity number 3333
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# profile gdoi-p
Device(gdoi-sa-ipsec)# match address ipv4 102
```

```
Device(gdoi-sa-ipsec)# end
```

Example: Configuring a Group Member for GET VPN Suite B

Configuring Ciphers or Hash Algorithms for the KEK for Suite B

The following example shows how to configure the Suite B ciphers and hash algorithms for the KEK to be allowed by the GM. This example uses an identity address (compatible only with IPv4 data plane configurations). You could instead use an identity number (which would be compatible with IPv4 and IPv6 data plane configurations).

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group gdoigroupone
Device(config-gdoi-group)# identity address ipv4 10.2.2.2
Device(config-gdoi-group)# server address ipv4 10.0.5.2
Device(config-gdoi-group)# client rekey encryption 3des-cbc aes 192 aes 256
Device(config-gdoi-group)# client rekey hash sha384
Device(config-gdoi-group)# end
```

Configuring Acceptable Transform Sets for TEKs for Suite B

The following example shows how to configure the acceptable transform sets used by TEKs for data encryption or authentication.

```
Device> enable
Device# configure terminal
Device(config)# crypto ipsec transform-set g1 esp-gcm 192
Device(cfg-crypto-trans)# exit
Device(config)# crypto gdoi group gdoigroupone
Device(config-gdoi-group)# client transform-sets g1
Device(config-gdoi-group)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS security commands	*Cisco IOS Security Command References*

Related Topic	Document Title
IKE and IKE policy configuration tasks IPsec transform configuration tasks	"Configuring Internet Key Exchange for IPsec VPNs" module in the Internet Key Exchange for IPsec VPNs Configuration Guide, Cisco IOS Release 15.2M&T
Basic deployment guidelines for enabling GET VPN in an enterprise network	Cisco IOS GET VPN Solutions Deployment Guide

Standards and RFCs

Standard/RFC	Title
Federal Information Processing Standard (FIPS) Publication 140-2	Security Requirements for Cryptographic Modules
RFC 2401	Security Architecture for the Internet Protocol
RFC 4106	The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)
RFC 4543	The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH
RFC 4869	Suite B Cryptographic Suites for IPsec
RFC 6054	Using Counter Modes with Encapsulating Security Payload (ESP) and Authentication Header (AH) to Protect Group Traffic
RFC 6407	The Group Domain of Interpretation

Technical Assistance

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

Feature Information for GET VPN Support with Suite B

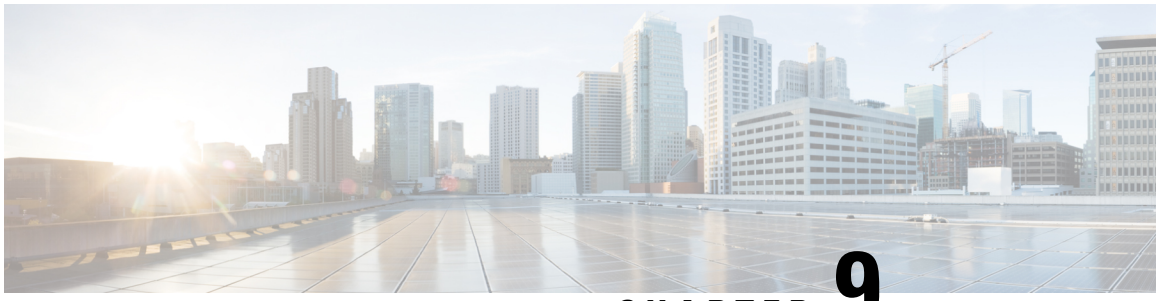
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train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

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

Table 18: Feature Information for GET VPN Support with Suite B

Feature Name	Releases	Feature Information
Ability to use Suite B Algorithms with GIKEv2 with registration interface	Cisco IOS XE Fuji 16.8.1	The Ability to use Suite B Algorithms with GIKEv2 with registration interface feature provides support for Suite B on GET VPN G-IKEv2 enabled networks. In Cisco IOS XE Fuji 16.8.1, this feature supported the following devices: • Cisco ASR 1000 Series Aggregation Services Routers • Cisco 1100 Series Integration Services Routers • Cisco 4000 Series Integrated Services Routers The following command was modified: **show crypto gdoi**.

Feature Name	Releases	Feature Information
GET VPN Support with Suite B	Cisco IOS XE Release 3.10S	The GET VPN Support with Suite B feature adds support of the Suite B set of ciphers to Cisco Group Encrypted Transport (GET) VPN. Suite B is a set of cryptographic algorithms that includes Galois Counter Mode Advanced Encryption Standard (GCM-AES) as well as algorithms for hashing, digital signatures, and key exchange. Suite B for IP security (IPsec) VPNs is a standard whose usage is defined in RFC 4869. Suite B provides a comprehensive security enhancement for Cisco IPsec VPNs, and it allows additional security for large-scale deployments. Suite B is the recommended solution for organizations requiring advanced encryption security for the wide-area network (WAN) between remote sites. The following commands were introduced or modified: **client rekey hash, crypto key export ec, crypto key generate ec keysize, crypto key import ec, group size, identifier, rekey sig-hash algorithm, show crypto gdoi.**



CHAPTER 9

GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

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

Finding Feature Information

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

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

Prerequisites for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

All key servers (KSs) and group members (GMs) on which you want to enable this feature must be running GET VPN software version 1.0.5 or higher. You should use this feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support it.

This feature provides a command that you use on the KS (or primary KS) to check whether all devices in the network are running versions that support IPsec inline tagging for Cisco TrustSec. For more information, see the "Ensuring That GMs Are Running Software Versions That Support IPsec Inline Tagging for Cisco TrustSec" section.

Restrictions for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

- This feature does not support IPv6 traffic.
- This feature does not support transport mode on the Cisco ASR 1000 Series Aggregation Services Routers or on the Cisco VPN Internal Service Module for Cisco Integrated Services Routers Generation 2 (ISR G2).

Information About GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

Group Member Registration of Security Group Tagging Capability

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 GET VPN 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.

Creation of SAs with Security Group Tagging Enabled

After you enable GET VPN support of IPsec inline tagging (using the **tag cts sgt** command) in a group SA and then trigger a rekey (using the **crypto gdoi ks rekey** command), the KS checks for GMs and cooperative KSs in the group not using a compatible software version. If found, a warning message appears:

```
WARNING for group GETVPN: some devices cannot support SGT inline tagging. Rekey can cause
traffic disruption and GM registration failures. Please check 'show crypto gdoi feature
sgt'.
Are you sure you want to proceed ? [yes/no]:
```

Handling of Security Group Tags in the Group Member Data Plane

Egress traffic is traffic sent out from a GDOI-protected interface of a GM. The following table specifies GM behavior for the egress path:

Table 19: Egress Handling of Security Group Tags

Security group tagging is enabled on SA	CTS provides SGTs	GM data plane behavior
Yes	Yes	Adds SGTs to Cisco metadata and encrypts
Yes	No	Encrypts without SGTs
No	Yes	Encrypts without SGTs
No	No	Encrypts without SGTs

Ingress traffic is traffic received by a GDOI-protected interface of a GM. The table below specifies GM behavior for the ingress path:

Table 20: Ingress Handling of Security Group Tags

Security group tagging is enabled on SA	CTS provides SGTs	GM data plane behavior
Yes	Yes	Decrypts and extracts SGTs for CTS
Yes	No	Decrypts without SGT processing
No	Yes	Decrypts and ignores SGTs
No	No	Decrypts without SGT processing

Packet Overhead and Fragmentation When Using Security Group Tagging

Because it adds Cisco metadata containing the SGT information to each GDOI packet, SGT inline tagging increases packet overhead by eight bytes (or 16 bytes with time-based antireplay enabled).

If a packet is fragmented before GDOI encryption, each fragment is inline tagged with SGT information accordingly. If packet is fragmented after GDOI encryption, only the first fragment is inline tagged with SGT information.

You can use two methods to handle fragmentation. The first method is to use the **ip mtu** command on the interface that is handling encryption to accommodate the extra bytes used to carry the SGT information via Cisco metadata. The second method is to use the **ip tcp adjst-mss 1352** command on the GM's LAN interface. This command ensures that the resulting IP packet on the LAN segment is less than 1392 bytes, thereby providing 108 bytes for any overhead plus the Cisco metadata to carry the SGTs.

For more information about designing around MTU issues, refer to the “Designing Around MTU Issues” section of the Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide

How to Configure GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

Ensuring That GMs Are Running Software Versions That Support IPsec Inline Tagging for Cisco TrustSec

You should use the IPsec Inline Tagging for Cisco TrustSec feature only after all devices in the GET VPN network are upgraded to GET VPN software versions that support this feature.

Perform this task on the KS (or primary KS) to ensure that all devices in the network support IPsec inline tagging for Cisco TrustSec.

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi feature cts-sgt**
3. **show crypto gdoi feature cts-sgt | include No**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show crypto gdoi feature cts-sgt** **Example:** `Device# show crypto gdoi feature cts-sgt`	Displays the version of the GET VPN software running on each KS and GM in the GET VPN network and displays whether that device supports IPsec inline tagging for Cisco TrustSec.
Step 3	**show crypto gdoi feature cts-sgt | include No** **Example:** `Device# show crypto gdoi feature cts-sgt | include No`	(Optional) Displays only those devices that do not support IPsec inline tagging for Cisco TrustSec.

Configuring IPsec Inline Tagging for Cisco TrustSec

To configure IPsec inline tagging for Cisco TrustSec, perform the following steps.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **crypto gdoi group** *group-name*
4. Enter one of the following commands:
   - **identity number** *number*
   - **identity address ipv4** *address*
5. **server local**
6. **sa ipsec** *sequence-number*
7. **tag cts sgt**
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GET-SGT`	Identifies a GDOI group and enters GDOI group configuration mode.

Step 4	Enter one of the following commands:
	• **identity number** *number*
	• **identity address ipv4** *address*
	Example:
	`Device(config-gdoi-group)# identity number 3333`
	Example:
	`Device(config-gdoi-group)# identity address ipv4 10.2.2.2`
		Identifies a GDOI group number or address.

Step 5	**server local** **Example:** `Device(config-gdoi-group)# server local`	Designates a device as a GDOI KS and enters GDOI local server configuration mode.

	Command or Action	Purpose
Step 6	**sa ipsec** *sequence-number* **Example:** `Device(gdoi-local-server)# sa ipsec 1`	Specifies the IPsec SA policy information to be used for a GDOI group and enters GDOI SA IPsec configuration mode.
Step 7	**tag cts sgt** **Example:** `Device(gdoi-sa-ipsec)# tag cts sgt`	Enables IPsec inline tagging for Cisco TrustSec.
Step 8	**end** **Example:** `Device(gdoi-sa-ipsec)# end`	Exits GDOI SA IPsec configuration mode and returns to privileged EXEC mode.

After enabling IPsec inline tagging, you must trigger a rekey. For more information, see the "Triggering a Rekey" section.

Triggering a Rekey

If you change the security policy (for example, from DES to AES) on the KS (or primary KS) and exit from global configuration mode, a syslog message appears on the KS indicating that the policy has changed and a rekey is needed. You enter the rekey triggering command as described below to send a rekey based on the latest policy in the running configuration.

Perform this task on the KS (or primary KS) to trigger a rekey.

SUMMARY STEPS

1. **enable**
2. **crypto gdoi ks** [**group** *group-name*] **rekey** [**replace-now**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**crypto gdoi ks** [**group** *group-name*] **rekey** [**replace-now**] **Example:** `Device# crypto gdoi ks group mygroup rekey`	Triggers a rekey on all GMs. The optional **replace-now** keyword immediately replaces the old TEKs and KEK on each GM to enable the new policy before the SAs expire. **Note** Using the **replace-now** keyword could cause a temporary traffic discontinuity.

Examples

A message appears on the KS as follows:

```
Device# crypto gdoi ks rekey
Device#
*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
```

After the policy change, when each GM receives this triggered rekey, it installs the new SAs (for example, for AES) and shortens the lifetimes of the old SAs (for example, for DES). Each GM continues to encrypt and decrypt traffic using the old SA until its shortened lifetime expires.

If you try to trigger a rekey on the secondary KS, it rejects the command as shown below:

```
Device# crypto gdoi ks rekey
ERROR for group GET: This command must be executed on Pri-KS
```

Verifying and Troubleshooting GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

To view the configuration that is running on a GM, use the **show running-config** command.

To display the number of packets that are tagged with SGTs, enter the following command.

```
Device# show crypto ipsec sa detail

interface: Ethernet0/0
   Crypto map tag: GET, local addr 5.0.0.2
   protected vrf: (none)
   local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
   Group: GET-SGT
.
.
.
#pkts tagged (send): 0, #pkts untagged (rcv): 5
```

The pkts tagged (send) field displays packets tagged with an SGT in the outbound direction. The pkts untagged (rcv) field displays packets not tagged with an SGT in the inbound direction.

Configuration Examples for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

Example: Ensuring That GMs Are Running Software Versions That Support IPsec Inline Tagging for Cisco TrustSec

The following example shows how to use the GET VPN 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 will display 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 GET VPN 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
```

Example: Configuring IPsec Inline Tagging for Cisco TrustSec

The following example shows how to configure CTS SGT inline tagging in an IPsec SA for a KS serving a single GDOI group:

```
Device> enable
Device# configure terminal
Device(config)# ip access-list extended ACL-SGT
Device(config-ext-nacl)# permit ip any any
Device(config-ext-nacl)# exit
Device(config)# crypto gdoi group GET-SGT
```

```
Device(config-gdoi-group)# identity number 1
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# tag cts sgt
Device(gdoi-sa-ipsec)# profile gdoi-p2
Device(gdoi-sa-ipsec)# match address ipv4 ACL-SGT
Device(gdoi-sa-ipsec)# replay time window-size 100
Device(gdoi-sa-ipsec)# end
```

The following example shows how to configure two groups: A group with GMs that are upgraded to GET VPN 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.

This example configures SGT tagging for traffic 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:

```
Device> enable
Device# configure terminal
Device(config)# ip access-list extended ACL_NET_AB
Device(config-ext-nacl)# permit ip 10.1.0.0 0.0.255.255 10.2.0.0 0.0.255.255
Device(config-ext-nacl)# permit ip 10.2.0.0 0.0.255.255 10.1.0.0 0.0.255.255
Device(config-ext-nacl)# exit
Device(config)# ip access-list extended ACL_ALL
Device(config-ext-nacl)# permit ip any any
Device(config-ext-nacl)# exit
Device(config)# crypto gdoi group GET1
Device(config-gdoi-group)# identity number 1111
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# rekey authentication mypubkey rsa mykey
Device(gdoi-local-server)# rekey transport unicast
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# tag cts sgt
Device(gdoi-sa-ipsec)# profile gdoi-p2
Device(gdoi-sa-ipsec)# match address ipv4 ACL_NET_AB
Device(gdoi-sa-ipsec)# replay time window-size 100
Device(gdoi-sa-ipsec)# exit
Device(gdoi-local-server)# exit
Device(config-gdoi-group)# exit
Device(config)# crypto gdoi group GET2
Device(config-gdoi-group)# crypto gdoi group GET2
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# rekey authentication mypubkey rsa mykey
Device(gdoi-local-server)# rekey transport unicast
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# profile gdoi-p2
Device(gdoi-sa-ipsec)# match address ipv4 ACL_ALL
Device(gdoi-sa-ipsec)# replay time window-size 100
Device(gdoi-sa-ipsec)# end
```

Note GET VPN supports a maximum of 100 ACEs per ACL.

Example: Triggering Rekeys on Group Members

Ensuring That GMs Are Running Software Versions That Support Rekey Triggering

The following example shows how to use the GET VPN software versioning command on the KS (or primary KS) to display the version of software on devices in the GET VPN network and display whether they support rekey triggering after a policy change:

```
Device# show crypto gdoi feature policy-replace

Key Server ID       Version  Feature Supported
  10.0.8.1         1.0.2          Yes
  10.0.9.1         1.0.2          Yes
  10.0.10.1        1.0.2          Yes
  10.0.11.1        1.0.2          Yes
 Group Member ID     Version  Feature Supported
  5.0.0.2          1.0.2          Yes
  9.0.0.2          1.0.1          No
```

The following example shows how to find only those devices that do not support rekey triggering after policy replacement:

```
Device# show crypto gdoi feature policy-replace  | include No

       9.0.0.2          1.0.1          No
```

For these devices, the primary KS sends only the triggered rekey without instructions for policy replacement. Therefore, when a GM receives the rekey, it installs the new SAs but does not shorten the lifetimes of the old SAs.

Triggering a Rekey

The following example shows how to trigger a rekey after you have performed a policy change. In this example, an IPsec policy change (for example, DES to AES) occurs with the **profile gdoi-p2** command:

```
Device# configure terminal
Device(config)# crypto gdoi group GET
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# sa ipsec 1
Device(gdoi-sa-ipsec)# no profile gdoi-p
Device(gdoi-sa-ipsec)# profile gdoi-p2
Device(gdoi-sa-ipsec)# end
Device#

*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
Device# crypto gdoi ks rekey
Device#
*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
```

The following example shows the error message that appears if you try to trigger a rekey on the secondary KS:

```
Device# crypto gdoi ks rekey

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

Note If time-based antireplay (TBAR) is set, the key server periodically sends a rekey to the group members every 2 hours (7200 sec). In the following example, even though the lifetime is set to 8 hours (28800 sec), the rekey timer is set to 2 hours.

```
Device(config)# crypto ipsec profile atm-profile
Device(ipsec-profile)# set security-association lifetime seconds 28800
!
Device(ipsec-profile)# exit
Device(config)# crypto gdoi group ATM-DSL
Device(config-gdoi-group)# server local
Device(gdoi-sa-ipsec)# sa ipsec 1
!
Device(gdoi-sa-ipsec)# replay time window-size 100
```

The commands **show crypto gdoi gm replay** and **show crypto gdoi ks replay** displays TBAR information.

Additional References for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS security commands	*Cisco IOS Security Command References*
Basic deployment guidelines for enabling GET VPN in an enterprise network	Cisco IOS GET VPN Solutions Deployment Guide
Configuring Cisco TrustSec	Cisco TrustSec Configuration Guide, Cisco IOS Release 15M&T
Designing around MTU issues	Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide

Standards and RFCs

Standard/RFC	Title
RFC 2401	Security Architecture for the Internet Protocol

Standard/RFC	Title
RFC 6407	The Group Domain of Interpretation

Technical Assistance

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

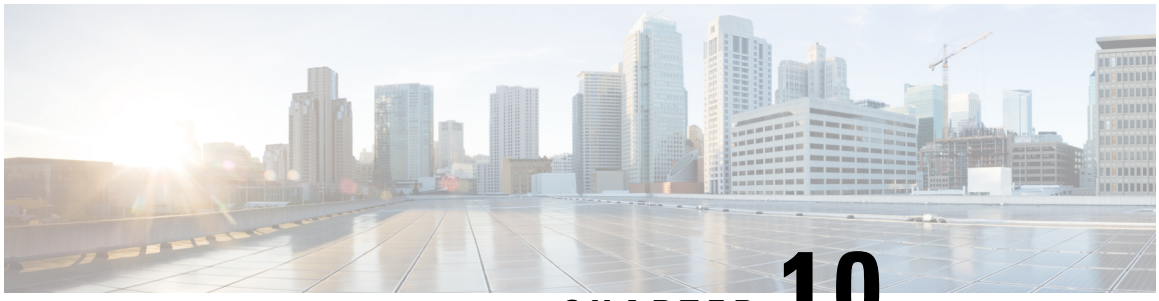
Feature Information for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

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

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

Table 21: Feature Information for GET VPN Support of IPsec Inline Tagging for Cisco TrustSec

Feature Name	Releases	Feature Information
GET VPN Support of IPsec Inline Tagging for Cisco TrustSec	Cisco IOS XE Release 3.9S	The 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 GET VPN Support of IPsec Inline Tagging for Cisco TrustSec feature uses GET VPN inline tagging to carry the SGT information across the private WAN. The following commands were introduced or modified: **show crypto gdoi, show crypto ipsec sa, tag cts sgt.**



CHAPTER 10

GETVPN GDOI Bypass

The GETVPN GDOI Bypass feature supports enabling and disabling the default Group Domain of Interpretation (GDOI) bypass crypto policy. It also supports hardening of the default GDOI bypass crypto policy once it is enabled.

Finding Feature Information

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

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

Restrictions for GETVPN GDOI Bypass

When a key server (KS) is placed behind a group member (GM), the local deny Access Control List (ACL) must be configured explicitly to allow traffic using UDP as the transport protocol and port 848 as either the source or destination (UDP 848 traffic) to pass through.

Information About GETVPN GDOI Bypass

GDOI Bypass Crypto Policy

The Cisco IOS Group Encrypted Transport VPN (GETVPN) uses Group Domain of Interpretation (GDOI) as the key management protocol.

A group member (GM) is a device responsible for encryption and decryption, that is, a device responsible for handling the GET VPN data plane.

A key server (KS) is a device responsible for creating and maintaining the GET VPN control plane. All encryption policies, such as 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.

Enabling and Disabling the Default GDOI Bypass Crypto Policy

A new group member (GM) configuration allows users to disable the Group Domain of Interpretation (GDOI) bypass crypto policy and to control traffic exceptions by explicitly configuring the GM local access control list (ACL).

Hardening of the Default GDOI Bypass Crypto Policy

To improve security, the following changes have been enforced while applying the default Group Domain of Interpretation (GDOI) bypass crypto policy:

- The default GDOI bypass crypto policy is installed only on Group Encrypted Transport VPN (GETVPN)-protected interfaces (interfaces at which GDOI crypto map is applied). Only UDP848 traffic that is destined for the group member's (GM) address used for registration or rekey is allowed.
- If the GM VRF-aware feature is used to specify that the GDOI data plane and control plane are in different VRFs, auto-insertion of the default GDOI bypass crypto policy is not applied to the GDOI-protected interface.
- If traffic using UDP as the transport protocol and port 848 as either the source or destination (UDP 848 traffic) is expected to arrive at other non-GDOI-protected interfaces (but with other crypto maps applied), exceptions for the non-GDOI crypto map must be explicitly configured.
- If a crypto map set with multiple groups is configured, the overall GDOI bypass crypto policy installed is the union of all the GDOI bypass crypto policies for each group within the security association database (SADB).

Any of the conditions mentioned below triggers a recompute of the default GDOI bypass crypto policy applied to a GETVPN-protected interface:

- Removing **client bypass-policy** configuration using the **no client bypass-policy** command.
- Applying or removing the GDOI bypass crypto map from an interface.
- Applying or removing the GDOI bypass crypto map from crypto map sets.
- Changing the IP address of the GDOI-protected interface (if **no client registration interface** is used)

- If **client registration interface** is used, the following cases trigger a recompute of the default GDOI bypass crypto policy applied to a GETVPN-protected interface:
  - Changes from **no client registration interface** to **client registration interface**
  - Changes to the client registration interface (for example, from loopback 0 to loopback 1)
  - Changes to the client registration interface address

How to Configure GETVPN GDOI Bypass

Enabling the Default GDOI Bypass Crypto Policy

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **client bypass-policy**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GETVPN`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**client bypass-policy** **Example:** `Device(config-gdoi-group)# client bypass-policy`	Enables the default GDOI bypass crypto policy.
Step 5	**end** **Example:** `Device(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Disabling the Default GDOI Bypass Crypto Policy

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **no client bypass-policy**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GETVPN`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**no client bypass-policy** **Example:** `Device(config-gdoi-group)# no client bypass-policy`	Disables the default GDOI bypass crypto policy.
Step 5	**end** **Example:** `Device(config-gdoi-group)# end`	Exits GDOI group configuration mode and returns to privileged EXEC mode.

Verifying Enablement and Disablement of the Default GDOI Bypass Crypto Policy

SUMMARY STEPS

1. **enable**
2. **show crypto gdoi gm acl**
3. **show crypto gdoi gm acl**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show crypto gdoi gm acl**

Verifies the enablement of the default GDOI bypass crypto policy.

Note VRF will be displayed only if it is non-global.

Example:

```
Device# 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*
```

Step 3 **show crypto gdoi gm acl**

Verifies the disablement of the default GDOI bypass crypto policy.

Example:

```
Device# 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: Disabled
```

Configuration Examples for GETVPN GDOI Bypass

Example: Enabling the Default GDOI Bypass Crypto Policy

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group getvpn
Device(config-gdoi-group)# client bypass-policy
Device(config-gdoi-group)# end
```

Example: Disabling the Default GDOI Bypass Crypto Policy

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group getvpn
Device(config-gdoi-group)# no client bypass-policy
Device(config-gdoi-group)# end
```

Additional References for GETVPN GDOI Bypass

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Cisco IOS security commands	*Cisco IOS Security Command References*
Basic deployment guidelines for enabling GET VPN in an enterprise network	*Cisco IOS GET VPN Solutions Deployment Guide*
Designing and implementing a GET VPN network	*Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide*

Standards and RFCs

Standard/RFC	Title
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

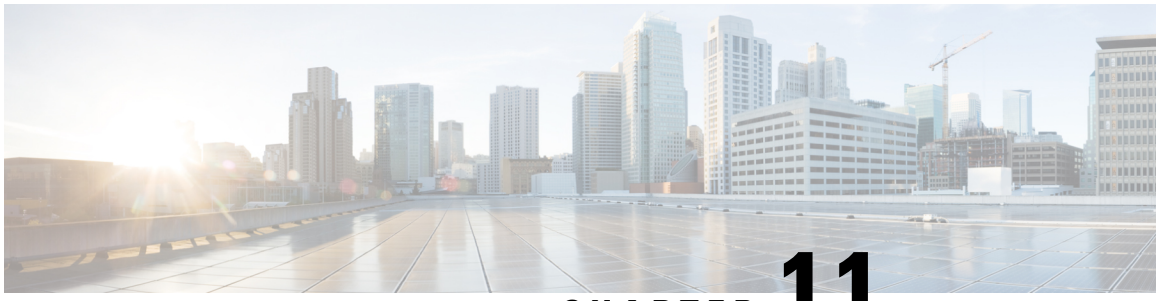
Feature Information for GETVPN GDOI Bypass

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

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

Table 22: Feature Information for GETVPN GDOI Bypass

Feature Name	Releases	Feature Information
GETVPN GDOI Bypass	Cisco IOS XE Release 3.13S	The GETVPN GDOI Bypass feature supports enabling and disabling the default Group Domain of Interpretation (GDOI) bypass crypto policy. It also supports hardening of the default GDOI bypass crypto policy once it is enabled. The following commands were introduced: **client bypass-policy** and **show crypto gdoi gm acl**.



CHAPTER 11

GETVPN G-IKEv2

Cisco Group Encrypted Transport VPN (GET VPN) includes a set of features that are necessary to secure IP multicast group traffic or unicast traffic over an enterprise private WAN that originates on or flows through a Cisco device. The GETVPN G-IKEv2 feature implements Internet Key Exchange version 2 (IKEv2) protocol on GETVPN thereby allowing GETVPN to derive the benefits of IKEv2.

Finding Feature Information

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

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

Restrictions for GETVPN G-IKEv2

- You can configure either Group Key Management (GKM) or Group Domain of Interpretation (GDOI) for a group member (GM), whereas you can configure both GKM and GDOI for a key server (KS).
- IKEv2 for COOP is not supported. Use IKEv1 for COOP between the key servers in the G-IKEv2 setup.
- EAP is not currently supported with G-IKEv2.
- GETVPN G-IKEv2 does not support IP-D3P. IP-D3P with G-IKEv2 is yet to be supported on GETVPN Group Members (GMs).

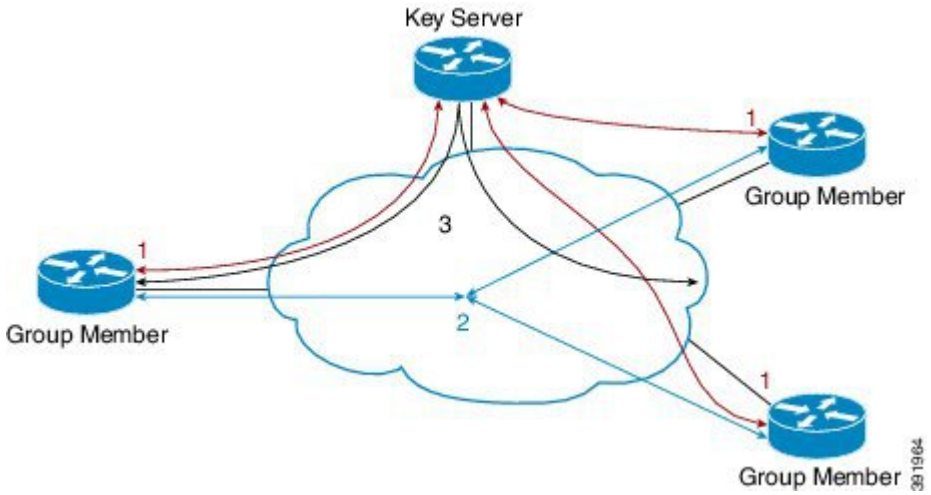
Information About GETVPN G-IKEv2

Overview of GETVPN G-IKEv2

Cisco Group Encrypted Transport Virtual Private Network (GETVPN) architecture is based on the Group Domain of Interpretation (GDOI) protocol. GETVPN uses Internet Security Exchange and Key Management Protocol (ISAKMP) to authenticate new group members, download cryptographic policy, and distribute traffic encryption key (TEK) and key encryption key (KEK) to group members. However, Internet Key Exchange Version 2 (IKEv2) has replaced. IKEv2 reduces network latency, reduces complexity in message exchanges, improves interoperability and reliability, and fixes cryptographic issue in HASH authentication. GET VPN 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. The following figure depicts the basic system architecture of GETVPN using G-IKEv2 to register GM's with a KS and download keys and policy to GM's from a KS.

Figure 16: GETVPN Architecture through G-IKEv2 Protocol



Internet Key Exchange Version 2 (IKEv2)

Internet Key Exchange Version 2 (IKEv2), a next-generation key management protocol based on RFC 4306, is an enhancement of the IKE Protocol. IKEv2 is used for performing mutual authentication and establishing and maintaining security associations (SAs). For more information on IKEv2, see *FlexVPN and Internet Key Exchange Version 2 Configuration Guide*.

The following table compares the tunnel performance between IKE and IKEv2.

Protocol	Tunnels per Second	Maximum Simultaneous Tunnels
IKE	45	60

Protocol	Tunnels per Second	Maximum Simultaneous Tunnels
IKEv2	89	200

The benefits of IKEv2 are as follows:

Dead Peer Detection and Network Address Translation-Traversal

Internet Key Exchange Version 2 (IKEv2) provides built-in support for Dead Peer Detection (DPD) and Network Address Translation-Traversal (NAT-T).

Certificate URLs

Certificates can be referenced through a URL and hash, instead of being sent within IKEv2 packets, to avoid fragmentation.

Denial of Service Attack Resilience

IKEv2 does not process a request until it determines the requester, which addresses to some extent the Denial of Service (DoS) problems in IKEv1, which can be spoofed into performing substantial cryptographic (expensive) processing from false locations.

Multiple Crypto Engines

If your network has both IPv4 and IPv6 traffic and you have multiple crypto engines, choose one of the following configuration options:

- One engine handles IPv4 traffic and the other engine handles IPv6 traffic.
- One engine handles both IPv4 and IPv6 traffic.

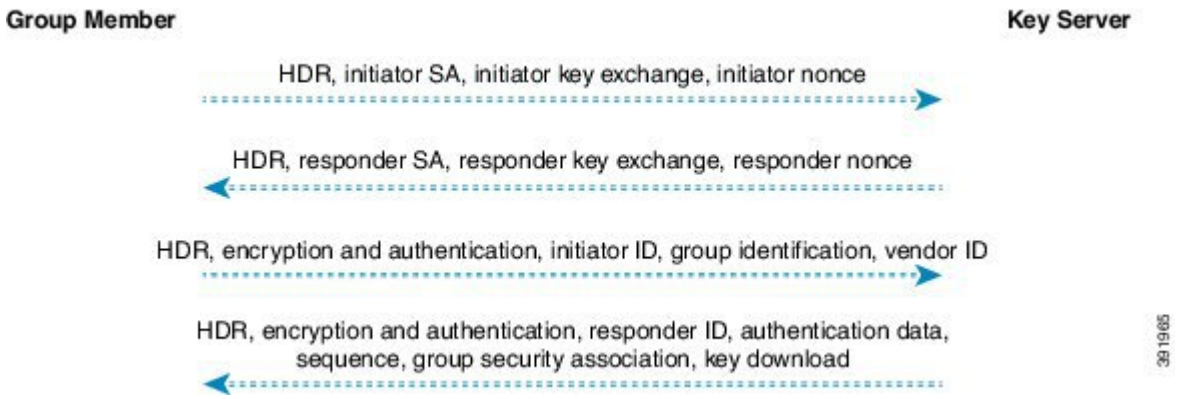
Reliability and State Management (Windowing)

IKEv2 uses sequence numbers and acknowledgments to provide reliability, and mandates some error-processing logistics and shared state management.

GETVPN G-IKEv2 Exchanges

The message exchanges between GM and KS conforms to the Internet Engineering Task Force (IETF) Group Key Management using IKEv2 Standards draft.

Figure 17: G-IKEv2 Message Exchanges



1. Group member initiates a registration request to key server by sending preferred cryptographic algorithms (in SAi payload), Diffie–Hellman public number, in initiator's key exchange (KE) phase 1 payload, and nonce, which is a random number for guaranteeing liveness in Initiator's nonce payload.
2. Key server responds with the negotiated cryptographic algorithm (in responder's SA phase 1 payload), Diffie–Hellman public number (in responder's KE payload), nonce (in responder's nonce payload). Optionally, if key server is configured to use Rivest, Shamir, and Adleman (RSA) digital signature as an authentication method, key server also sends a certificate request.
3. On receiving key server's response to the registration request, the group member uses the cryptographic algorithm in the SAr1 payload and Diffie–Hellman value to create keys and to encrypt the message sent to the key server. The encrypted message includes the initiator's ID and, optionally, certificate and certificate request, if RSA digital signature is used as authentication method. In case of Suite B implementations, a notify payload is sent for requesting sender IDs used with Galois/Counter Mode (GCM)–Advanced Encryption Standard (AES) or Galois Message Authentication Code (GMAC)–Advanced Encryption Standard (AES) transforms.

Note Group member requests a set of sender IDs applicable for interfaces for a lifetime of one day. After receiving the lifetime in a registration (for Long SA Lifetime) or a rekey (for Short SA Lifetime) message, group member stores the lifetime for calculating the number of sender IDs for future registrations.

4. After authenticating group manager, key server authorizes group member before registering group manager. After registration, key server sends the group's policy (in the GSA payload) and the group's keying material (in the KD payload) to group manager. The SEQ payload is optional and is sent when the key server wants to inform group manager of the current sequence number of the rekey message. These payloads are included in the GSA_AUTH response message.

Group Member Communication

Group members do not establish IPsec tunnels with one another, but use the IPsec policy and keys to secure communication between group members in a group.

Future Registrations

When a secure registration channel is established between group manager and key server, additional group member registrations for additional groups occurs through the established secure registration channel. In such scenario, group member uses the GSA_CLIENT_SERVER exchange that includes the group ID (IDg) to request either key encryption keys (KEK) or traffic encryption keys (TEKs) or both from key server.

Key Server Rekey

Key server distributes new group keys to group members using the G-IKEv2 group maintenance channel via unicast or multicast communication. Rekey is optional in G-IKEv2. When rekey is used, the KS sends a rekey message to group member. This message could be unicast or multicast depending on the key server configuration. Key server uses the KEK that is sent to the group member during registration to encrypt the rekey message. On receiving a rekey message, group member must ensure that the SEQ number in the rekey message is larger than the last received SEQ number. Group member could have received the SEQ number either via a registration message or a rekey message, whichever is later. If key server group is configured as both GDOI (IKEv1) and G-IKEv2 group, two rekey messages are sent—one over GDOI and another over G-IKEv2—for multicast rekey. In case of unicast rekey, key server only sends a GDOI or G-IKEv2 rekey depending on the group member's mode or type.

Note If the rekey is unicast, the group member must send an acknowledgment to key server.

Supported Features and GKM Version

The GETVPN G-IKEv2 feature supports the existing GETVPN features, which are as follows:

- Rekey and retransmission
- GM access control list (ACL)
- Fail-close mode
- Receive-only mode
- Anti-replay
- Authentication policy for group member registration
- GDOI MIBS
- VRF-Aware group member
- Group member removal and policy replacement
- Cooperative key server
- GETVPN IPv6 dataplane
- IPsec inline tagging support
- GETVPN resiliency phase 1 and phase 2
- Cooperative announcement message optimization

The GETVPN G-IKEv2 feature is supported in GKM version 1.0.12 and later releases. The supported GKM versions for a key server is 1.0.13 and a group member is 1.0.12. The difference between versions on a key server and a group member is because the IP D3P support on GETVPN Key Server and Internet-Draft ACK for Cisco GETVPN Key Server features are available on the key server from 1.0.13 only.

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. 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.

Service Upgrade Procedure

1. Save the existing key server and group member GDOI configurations. For more information, see the "Configuration Replace and Configuration Rollback" feature module in the *Managing Configuration Files Configuration Guide*.
2. Configure a key encryption key (KEK) and a traffic encryption key (TEK) lifetime on all key servers to avoid network split and merge during the migration of the key servers. Use the crypto gdoi ks rekey command to configure the new lifetimes.
3. Upgrade key server to the new Cisco IOS software images. Follow the sequence mentioned above—start with the secondary key server followed by the primary key server. All existing configurations that use the keyword **gdoi** will be converted to the keyword **gkm**. For example, the global configuration command **crypto gdoi group** will be converted to **crypto gkm group** command. However, the groups continue to use GDOI for registration and rekey.
4. On key server, execute the **gikev2** command in the server local command for groups that support GDOI and G-IKEv2 group members.
5. Upgrade group members to the new Cisco IOS software image. All existing configurations that use the keyword "**gdoi**" will be converted to the keyword **gkm**. For example, the global configuration commands **crypto gdoi group** and **crypto map gdoi** will be converted to "**crypto gkm group**" and **crypto map gkm** respectively. These groups continue to use GDOI for registration and rekey and include the **client protocol gdoi** command.
6. Configure the **client protocol gikev2** command to use G-IKEv2 on group member.
7. Configure the **no gdoi** command in the server local command, to stop servicing GDOI group members.

For a group member to use GDOI after upgrading to G-IKEv2, configure the **client protocol gdoi** command in the group member group configuration. Group member registers again with key server using GDOI instead of G-IKEv2.

Note Before you convert group member, ensure that key server to which group member is registered is configured with the gdoi command in GDOI local server configuration mode.

Service Downgrade Procedure

Use the previously saved GDOI configurations (saved before upgrade procedure) and downgrade the Cisco IOS software for each group member. Next, downgrade the key server; beginning with the secondary key server followed by primary key server. For more information, see the "Configuration Replace and Configuration Rollback" feature module in the *Managing Configuration Files Configuration Guide*.

Migration Examples

This section provides examples on GDOI to G-IKEv2 migration. The following examples show how the GDOI group g1 is converted to a GKM group after upgrading to a G-IKEv2 Cisco IOS software image. The following is a sample key server configuration before Cisco IOS software upgrade.

```
crypto gdoi group g1
 identity 1111
 server local
  .
  .
  .
  sa ipsec 1
   profile getvpn_profile
   match address getvpn_acl
   .
   .
   .
   redundancy
   .
   .
   .
```

The following is a sample key server configuration after Cisco IOS software upgrade. In this example, the commands **gdoi**, **no gikev2**, and **gikev2** are automatically added. The **gikev2** command starts accepting G-IKEv2 registrations.

```
crypto gkm group g1
 identity 1111
 server local
  gdoi
  no gikev2
  gikev2 ikev2_profile1
  .
  .
  .
  sa ipsec 1
   profile getvpn_profile
   match address getvpn_acl
   .
   .
   .
   redundancy
```

.
.
.

The following is a sample group member configuration before Cisco IOS software upgrade.

```
crypto gdoi group g1
 identity 1111
 server address ipv4 ks1
 server address ipv4 ks2

crypto map GETVPN_CM 10 gdoi
 set group g1

interface g0/0/0
 crypto map GETVPN_CM
```

The following is a sample group member configuration after Cisco IOS software upgrade. In this example, the commands **client protocol gdoi** and **client protocol gikev2** are automatically added. The **client protocol gikev2** command starts using G-IKEv2.

```
crypto gkm group g1
 identity 1111
 server address ipv4 ks1
 server address ipv4 ks2
 client protocol gdoi
 client protocol gikev2 ikev2_profile1 ] – Configure this to start using G-IKEv2

crypto map GETVPN_CM 10 gkm
 set group g1

interface g0/0/0
 crypto map GETVPN_CM
```

GETVPN G-IKEv2 Configuration

All GETVPN commands—EXEC and global configuration commands—include the keyword **gdoi**. 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. As of now, both commands **crypto gdoi** and **crypto gkm** are available. However, the **GDOI** keyword will be deprecated and replaced by the **gkm** keyword in future. For example, to configure a key server group, the GDOI command is **crypto gdoi group** *group-name*, whereas the GKM command would be **crypto gkm group** *group-name*.

G-IKEv2 Enhancement for GETVPN

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) and distinguished name (DN). The identity of a GM used for authorization is same as the identity used in IKEv2 protocol. In addition to the above listed identities, IKEv2 also supports email and key-id. When a GM registers to a KS using G-IKEv2, the registration fails when GM uses email or key-id as identities. Secondly, the KS is authorized twice for a GM—by IKEv2 and group key management (GKM) respectively.

The 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.

On G-IKEv2 networks, there are no additional commands required to configure IKEv2 profile-based authorization. The authorization is taken care when configuring the IKEv2 profile. You must only attach the IKEv2 profile to a GKM group (as a part of Get VPN G-IKEv2 feature) to the KS and the GM. G-IKEv2 registration fails if the same IKEv2 profile is not attached to both KS and GM. The G-IKEv2 Enhancement for GETVPN feature allows you to use the additional identities supported by IKEv2 on G-IKEv2 networks.

In case of GDOI-based networks, the authorization must be configured via the **authorization** command.

Note Although, GDOI and G-IKEV2 support access control list configuration in a 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.

How to Configure GETVPN G-IKEv2

Configuring an IKEv2 Profile

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 profile** *profile-name*
4. **authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}}
5. **identity local** {**address** {*ipv4-address* | *ipv6-address*} | **dn** | **email** *email-string* | **fqdn** *fqdn-string* | **key-id** *opaque-string*}
6. **keyring** {**local** *keyring-name* | **aaa** *list-name* [**name-mangler** *mangler-name* | **password** *password*] }
7. **match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map* | **fvrf** {*fvrf-name* | **any**} | **identity remote address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | {**email** [*domain string*] | **fqdn** [*domain string*]} *string* | **key-id** *opaque-string*}
8. **pki trustpoint** *trustpoint-label* [**sign** | **verify**]
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile gkm-gikev2`	Defines an IKEv2 profile and enters IKEv2 profile configuration mode.
Step 4	**authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}} **Example:** `Device(config-ikev2-profile)# authentication local ecdsa-sig`	Specifies the local or remote authentication method. • **rsa-sig** —Specifies RSA-sig as the authentication method. • **pre-share** —Specifies the preshared key as the authentication method. • **ecdsa-sig** —Specifies ECDSA-sig as the authentication method. • **eap** —Specifies EAP as the remote authentication method. • **query-identity** —Queries the EAP identity from the peer. • **timeout** *seconds* —Specifies the duration, in seconds, to wait for the next IKE_AUTH request after sending the first IKE_AUTH response. **Note** You can specify only one local authentication method but multiple remote authentication methods.
Step 5	**identity local** {**address** {*ipv4-address* | *ipv6-address*} | **dn** | **email** *email-string* | **fqdn** *fqdn-string* | **key-id** *opaque-string*} **Example:** `Device(config-ikev2-profile)# identity local email abc@example.com`	(Optional) Specifies the local IKEv2 identity type. **Note** If the local authentication method is a preshared key, the default local identity is the IP address. If the local authentication method is a Rivest, Shamir, and Adleman (RSA) signature, the default local identity is a Distinguished Name.
Step 6	**keyring** {**local** *keyring-name* | **aaa** *list-name* [**name-mangler** *mangler-name* | **password** *password*] } **Example:** `Device(config-ikev2-profile)# keyring aaa keyring1 name-mangler mangler1`	Specifies the local or AAA-based key ring that must be used with the local and remote preshared key authentication method. **Note** You can specify only one key ring. Local AAA is not supported for AAA-based preshared keys. **Note** Depending on your release, the **local** keyword and the **name-mangler** *mangler-name* keyword-argument pair should be used.

	Command or Action	Purpose
		Note When using AAA, the default password for a Radius access request is "cisco". You can use the **password** keyword within the **keyring** command to change the password.
Step 7	**match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map* | **fvrf** {*fvrf-name* | **any**} | **identity remote address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | {**email** [*domain string*] | **fqdn** [*domain string*]} *string* | **key-id** *opaque-string*} **Example:** `Device(config-ikev2-profile)# match address local interface Ethernet 2/0`	Uses match statements to select an IKEv2 profile for a peer.
Step 8	**pki trustpoint** *trustpoint-label* [**sign** | **verify**] **Example:** `Device(config-ikev2-profile)# pki trustpoint tsp1 sign`	Specifies Public Key Infrastructure (PKI) trustpoints for use with the RSA signature authentication method. **Note** If the **sign** or **verify** keyword is not specified, the trustpoint is used for signing and verification. **Note** In contrast to IKEv1, a trustpoint must be configured in an IKEv2 profile for certificate-based authentication to succeed. There is no fallback for globally configured trustpoints if this command is not present in the configuration. The trustpoint configuration applies to the IKEv2 initiator and responder.
Step 9	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits IKEv2 profile configuration mode and returns to privileged EXEC mode.

Configuring GKM Policy on a Key Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group** [**ipv6**] *group-name*
4. **server local**
5. **gikev2** *IKEv2-profile-name*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable**	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: `Device> enable`	• Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gkm group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gkm group gkm-grp1`	Configures a GKM policy and enters GKM group configuration mode.
Step 4	**server local** **Example:** `Device(config-gkm-group)# server local`	Designates a device as a GKM key server and enters GKM local server configuration mode.
Step 5	**gikev2** *IKEv2-profile-name* **Example:** `Device(gkm-local-server)# gikev2 gkm-gikev2`	Enables G-IKEv2 profile for registration and rekey on a key server.
Step 6	**end** **Example:** `Device(gkm-local-server)# end`	Exits GKM local server configuration mode and returns to privileged EXEC mode.

Configuring GKM Policy on Group Member

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group** [**ipv6**] *group-name*
4. **client protocol gikev2** *gkm-gikev2*
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**crypto gkm group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gkm group gkm-grp2`	Configures a GKM policy and enters GKM group configuration mode.
Step 4	**client protocol gikev2** *gkm-gikev2* **Example:** `Device(config-gkm-group)# client protocol gikev2 gkm-gikev2`	Enables G-IKEv2 profile for registration and rekey on a group member.
Step 5	**end** **Example:** `Device(config-gkm-group)# end`	Exits GKM group configuration mode and returns to privileged EXEC mode.

Configuring Authorization for GDOI Networks

This authorization is applicable to GDOI networks only. IKEv2 performs the authorization for G-IKEv2 networks.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group** [**ipv6**] *group-name*
4. **server local**
5. **authorization** {**address ipv4** {*acl-number* | *acl-name*} | **identity** *identity-name*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gkm group** [**ipv6**] *group-name* **Example:** `Device(config)# crypto gkm group gkm-grp1`	Configures a GKM policy and enters GKM group configuration mode.

	Command or Action	Purpose
Step 4	**server local** **Example:** `Device(config-gkm-group)# server local`	Designates a device as a GKM key server and enters GKM local server configuration mode.
Step 5	**authorization** {**address ipv4** {*acl-number* | *acl-name*} | **identity** *identity-name*} **Example:** `Device(gkm-local-server)# authorization identity email example@abc.com`	Configures authorization for the GKM group. • **address ipv4**—Specifies authorization by IPv4 address. • *acl-number*—The access control list number. Range: 1 to 99. • *acl-name*—The access control list number. • **identity** *identity-name*—Specifies authorization by identity, which can be one of the following: email, fully qualified domain name, or key-id.
Step 6	**end** **Example:** `Device(gkm-local-server)# end`	Exits GKM local server configuration mode and returns to privileged EXEC mode.

Additional References for GETVPN G-IKEv2

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security Commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*

Standards and RFCs

Standard/RFC	Title
RFC 4306	*Internet Key Exchange (IKEv2) Protocol*
Group Key Management using IKEv2	*draft-yeung-g-ikev2-07*

Technical Assistance

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

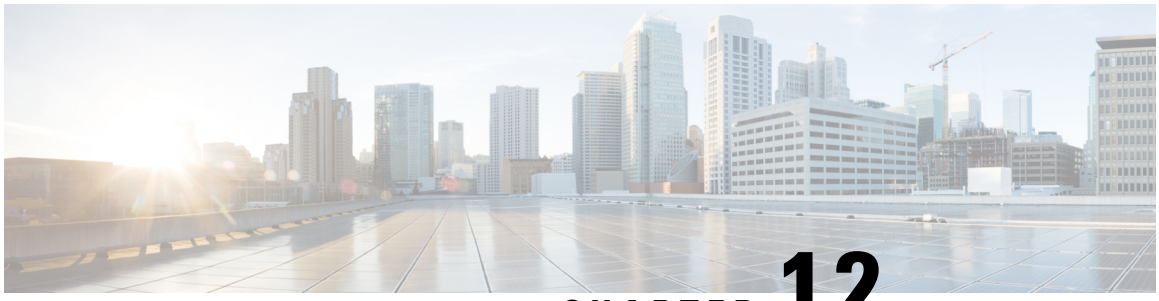
Feature Information for GETVPN G-IKEv2

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

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

Table 23: Feature Information for GETVPN G-IKEv2

Feature Name	Releases	Feature Information
GETVPN G-IKEv2	Cisco IOS XE Release 3.14S	Cisco Group Encrypted Transport VPN (GET VPN) includes a set of features that are necessary to secure IP multicast group traffic or unicast traffic over an enterprise private WAN that originates on or flows through a Cisco device. The GETVPN G-IKEv2 feature implements Internet Key Exchange version 2 (IKEv2) protocol on GETVPN thereby allowing GETVPN to derive the benefits of IKEv2. In Cisco IOS XE 3.14S, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers and Cisco 4400 Series Integrated Services Routers. The following commands were introduced or modified: **client protocol**, **crypto gkm group**, **gikev2**, **show crypto gkm**.
G-IKEv2 enhancement for GETVPN	Cisco IOS XE Fuji 16.8.1	The G-IKEv2 Enhancement for GETVPN feature leverages IKEv2 supported identities thereby enabling IKEv2 profile-based authorization on G-IKEv2 networks. In Cisco IOS XE Fuji 16.8.1, this feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. The following commands were introduced or modified: **authorization**, **show crypto gkm**.



CHAPTER 12

8K GM Scale Improvement

The 8K GM Scale Improvement feature supports optimization of the Cooperative Protocol (COOP) announcement messages by increasing the number of Group Members (GM) to 8000.

Finding Feature Information

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

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

Prerequisites for 8K GM Scale Improvement

To upgrade or downgrade a particular protocol version, maintain the same policies, keys, and group member (GM) database to ensure uninterrupted communication between GMs.

Information About 8K GM Scale Improvement

8K GM Scale Improvement

A Cooperative Protocol Announcement (COOP ANN) message has several clients and each client is associated with a protocol version. The COOP ANN message has been optimized to hold up to 8000 Group Members (GM), subsequently increasing the protocol version of the GM header.

This feature also supports upgrade and downgrade of a GM header protocol version.

How to Configure 8K GM Scale Improvement

Upgrading and Downgrading the Group Member Header Protocol Version

Before you begin

- Ensure that all Key Servers (KS) are upgraded to the "optimize" protocol version before scaling the network to more than 4000 GMs
- Ensure that all upgraded KSs must be downgraded to the "base" protocol version before scaling down to a network that supports only up to 4000 GMs.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gdoi group** *group-name*
4. **server local**
5. **redundancy**
6. **protocol version** {**base** | **optimize**}
7. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**crypto gdoi group** *group-name* **Example:** `Device(config)# crypto gdoi group GETVPN`	Identifies a GDOI group and enters GDOI group configuration mode.
Step 4	**server local** **Example:** `Device(config-gdoi-group)# server local`	Identifies a group server defined locally and enters GDOI local server configuration mode.
Step 5	**redundancy** **Example:** `Device(gdoi-local-server)# redundancy`	Enters GDOI COOP KS configuration mode. **Note** Ensure that the local server source address is defined.
Step 6	**protocol version** {**base** | **optimize**} **Example:** `Device(gdoi-coop-ks-config)# protocol version optimize`	Upgrades or downgrades the protocol version of the GM header. • **base**—COOP ANN message supports up to 4000 GMs. • **optimize**—COOP ANN message supports up to 8000 GMs.
Step 7	**end** **Example:** `Device(gdoi-coop-ks-config)# end`	Exits COOP KS configuration mode and returns to privileged EXEC mode.

Configuration Examples for 8K GM Scale Improvement

Example: Upgrading the Group Member Header Protocol Version

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group getvpn
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# redundancy
Device(gdoi-coop-ks-config)# protocol version optimize
Device(gdoi-coop-ks-config)# end
```

Example: Downgrading the Group Member Header Protocol Version

```
Device> enable
Device# configure terminal
Device(config)# crypto gdoi group getvpn
Device(config-gdoi-group)# server local
Device(gdoi-local-server)# redundancy
Device(gdoi-coop-ks-config)# protocol version base
Device(gdoi-coop-ks-config)# end
```

IPSEC Encryption and Decryption in GETVPN

In GETVPN IPsec flow, inbound traffic decryption might not happen in the expected IPsec flow recorder. The decrypted traffic can be recorded in any IPsec SA, if available. The decryption might happen in a random IPsec flow recorder. The following is an example:

```
Device# ping vrf cust1 48.1.1.1 so 38.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 48.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 38.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms

Device# show crypto session ivrf cust1  detail | sec permit ip 38.0.0.0
  IPSEC FLOW: permit ip 38.0.0.0/255.0.0.0 48.0.0.0/255.0.0.0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 16
mins
        Outbound: #pkts enc'ed 5 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 16
mins

Device# show crypto session ivrf cust1  detail | sec permit ip 48.0.0.0
  IPSEC FLOW: permit ip 48.0.0.0/255.0.0.0 38.0.0.0/255.0.0.0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 0 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 16
mins
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 16
mins

Device# show crypto session ivrf cust1  detail | sec permit ip 45.0.0.0
  IPSEC FLOW: permit ip 45.0.0.0/255.0.0.0 35.0.0.0/255.0.0.0
        Active SAs: 2, origin: crypto map
        Inbound:  #pkts dec'ed 5 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 15
mins
        Outbound: #pkts enc'ed 0 drop 0 life (KB/Sec) KB Vol Rekey Disabled/1 hours, 15
mins
```

In the above example, flow inbound traffic is not decrypted in the expected IPsec flow.

To overcome this issue and view the number of encrypted and decrypted packets, you can use the following **show** command. Here's a sample output of the **show** command.

```
Device# show crypto gdoi group v6-cust-gdoi1 gm dataplane counters

Data-plane statistics for group v6-cust-gdoi1:
   #pkts encrypt             : 1912   #pkts decrypt             : 1914
    #pkts tagged (send)      : 1841    #pkts untagged (rcv)     : 1834
    #pkts no sa (send)       : 0       #pkts invalid sa (rcv)   : 0
    #pkts encaps fail (send) : 0       #pkts decap fail (rcv)   : 0
    #pkts invalid prot (rcv) : 0       #pkts verify fail (rcv)  : 0
    #pkts not tagged (send)  : 0       #pkts not untagged (rcv) : 0
    #pkts internal err (send): 0       #pkts internal err (rcv) : 0
```

Additional References for 8K GM Scale Improvement

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Cisco IOS security commands	*Cisco IOS Security Command References*
Basic deployment guidelines for enabling GET VPN in an enterprise network	*Cisco IOS GET VPN Solutions Deployment Guide*
Designing and implementing a GET VPN network	*Group Encrypted Transport VPN (GETVPN) Design and Implementation Guide*

Standards and RFCs

Standard/RFC	Title
RFC 6407	*The Group Domain of Interpretation*

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

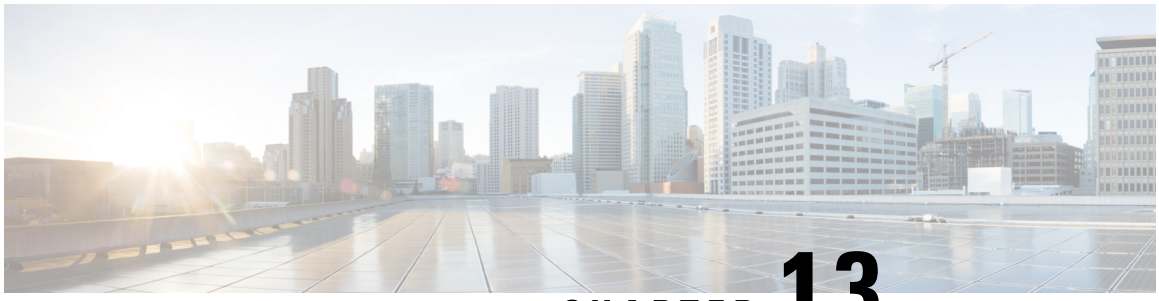
Feature Information for 8K GM Scale Improvement

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

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

Table 24: Feature Information for 8K GM Scale Improvement

Feature Name	Releases	Feature Information
8K GM Scale Improvement	Cisco IOS XE Release 3.14S	The 8K GM Scale Improvement feature supports optimization of the Cooperative Protocol (COOP) announcement messages by increasing the number of Group Members (GM) to 8000. The following command was modified: **protocol**.



CHAPTER 13

GET VPN Interoperability

The D3P Support on GETVPN Key Server, Activation Time Delay, and GDOI Interop ACK for Cisco GETVPN Key Server features enhance interoperability between key servers and group members.

Prerequisites for GET VPN Interoperability

- To enable the feature for a group, ensure that all the devices in a group are running compatible Cisco IOS software and Group Domain of Interpretation (GDOI) versions.
- Enable the Unicast Rekey functionality on a GDOI group before configuring the Internet-Draft ACK for Cisco GETVPN Key Server and Activation Time Delay features.

Restrictions for GET VPN Interoperability

- The IP-D3P support on GETVPN Key Server feature cannot coexist with the GETVPN Resiliency - GM Error Detection and GET VPN Support of IPsec Inline Tagging for Cisco TrustSec features. The latter features must be disabled before enabling IP-D3P support on a GET VPN key server and the IP-D3P must be disabled before enabling GETVPN Resiliency support on GETVPN Key Server.
- The Activation Time Delay feature supports only on IPsec security association. Multiple IPsec SA must not be configured.
- Cisco-Metdata and IP-D3P cannot coexist. When switching between CMD-feature and IP-D3P, the keyserver must perform **crypto gdoi ks rekey replace** to all the GMs to make sure these two features are not enabled simultaneously.
- ASR1K supports IP-D3P only in GETVPN IPv4 tunnel mode.

Information About GET VPN Interoperability

Overview of IP-Delivery Delay Detection Protocol (IP-D3P)

IP datagrams can be subject to a delivery delay attack, where a host or gateway receives datagrams that are not fresh. A fresh datagram is defined as a "Recently generated; not replayed from some earlier interaction of the protocol." An IP-D3P datagram consists of a header and an IP payload. The IP-D3P header includes a timestamp that is used by the receivers of the packet to determine if the packet has been recently generated. Receivers compare the timestamp delivered in the IP packet to their local time and thus determine whether the packet should be accepted.

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) to synchronize the system clocks of the sender and receiver.

The D3P support on GETVPN Key Server feature enables support for IP-D3P on GET VPN.

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. After you enable the D3P command, the primary key server issues a rekey to all the group members having a Group Associated Policy (GAP) payload with D3P attributes. The GAP payload includes the following attributes in the rekey message:

- D3P-TYPE—Portable Operating System Interface (POSIX) time, in milliseconds.
- D3P-WINDOWSIZE—IP-D3P window size, in milliseconds.

The **show crypto gkm ks** command displays the IP-D3P parameters that are enabled on a key server.

IP-D3P Support for Cooperative Key Server

If a GET VPN group has more than one key server, IP-D3P must be enabled on all the key servers. The primary key server sends the GAP payload containing the IP-D3P attributes to the secondary key servers through an announcement message, which notifies all cooperative key servers that IP-D3P is now enforced in the group.

On receiving the GAP payload, cooperative key servers check the IP-D3P attributes against their group configuration. If there is a mismatch, cooperative key servers generate a syslog message, warning the network administrator of a misconfiguration or incorrect configuration, as:

```
%GDOI-3-COOP_CONFIG_MISMATCH: WARNING: IP-D3P configuration between Primary KS and Secondary
 KS are mismatched
```

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. For example, if a key server configuration is **d3p window msec 1000** and a group member configuration is **client d3p window sec 50**, the group member can enable IP-D3P using the following parameters and overriding the parameters received from thekey server:

```
D3P-TYPE = POSIX-TIME-MSEC
D3P-WINDOWSIZE = 50000
```

Use the **show crypto gdoi gm** command to display the IP-D3P configuration of a group member and the IP-D3P errors, if any, that were encountered.

Note IP-D3P cannot be enabled on Cisco ASR 9000 Series Aggregation Services Routers, which use the parameters sent by a key server. Use the **show crypto gdoi group** command to view the parameters sent by the key server on Cisco ASR 9000 Series Aggregation Services Routers.

Activation Time Delay

GET VPN supports the Activation Time Delay (ATD) feature, in which a key server instructs group members to delay the use of new security associations (SAs) for traffic encryption. A key server includes the ATD value in the Group Associated Policy (GAP) payload when sending unicast rekey messages to group members. The time delay value is not user configurable; it is fixed as 30 seconds before SA expiry. The formula for calculating the ATD value is as follows:

ATD = Max((Max(old-SA-remaining-lifetime_sec, 30sec) – 30sec), 1sec)

Note ATD support is limited to group members that are configured on Cisco ASR 9000 Series Aggregation Services Routers and on non-Cisco devices. Therefore, a key server does not send ATD information to devices other than Cisco ASR 9000 Series Aggregation Services Routers and non-Cisco devices

.

Rekey Acknowledgment

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.

Cisco Unicast Rekey Acknowledgment Message

If a unicast rekey is configured, a key server sends rekey messages, for which group members reciprocate by sending an acknowledgment rekey message.

Note There is no acknowledgment message if multicast rekey is configured.

If a key server sends three consecutive unacknowledged unicast rekeys to a group member, and if the unicast rekeys are unacknowledged by that group member, the group member is removed from the group member database in the key server and no further unicast rekeys are sent to that group member.

GDOI I-D Rekey Acknowledgement Message

The GDOI Interop ACK for Cisco Key Server feature implements the standards for rekey acknowledgment messages between non-Cisco group members and a key server, as defined in the RFC-8263, GROUPKEY-PUSH Acknowledgment message.

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.

GDOI I-D Rekey ACK Support for a Key Server

The **rekey acknowledgement** command enables the key server to request group members to acknowledge rekeys depending on the keywords chosen with the command:

- **cisco**—Accepts Cisco-proprietary rekey ACK (encrypted) message.
- **interoperable**—Requests and accepts rekey ACK (unencrypted) message as per the corresponding Internet Draft.
- **any**—Accepts any supported ACK message based on the group key member version.

After enabling the **rekey acknowledgement** command, the key server sends a new policy attribute, **KEK_ACK_REQUESTED**. The new policy attribute in the key encryption key (KEK) SA payload for registration and rekey.

GDOI I-D Rekey ACK Support for Cooperative Key Server

The **rekey acknowledgement** command must be configured on all the key servers if a GET VPN group has multiple key servers. When a primary key server sends an announcement message to a secondary key server, the primary key server also includes the **KEK SA** payload carrying the **KEK_ACK_REQUESTED** attribute. This notifies all the cooperative key servers to send the **KEK_ACK_REQUESTED** attribute to the group members registered under them.

Upon receiving the **KEK SA** payload with the **KEK_ACK_REQUESTED** attribute, cooperative key servers check their group configuration. If there is a mismatch, cooperative key servers generate a message, warning the network administrator of a misconfiguration or incorrect configuration, as shown here:

```
%GDOI-3-COOP_CONFIG_MISMATCH: WARNING: Interoperable Rekey ACK configuration between Primary
 KS and Secondary KS are mismatched
```

Note Rekey acknowledgments are sent only to a primary key server because it is the primary key server that sends rekey messages. A rekey acknowledgment is sent to a cooperative server only when a cooperative key server is promoted as a primary key server, and if the old primary key server did not create a key encryption key (KEK) or traffic encryption key (TEK) policy.

GDOI I-D Rekey Support for Group Member

A group member is said to support the Internet-Draft ACK for Cisco GETVPN Key Server feature if the group member receives the rekey message containing the KEK_ACK_REQUESTED attribute in the KEK SA payload and sends the GDOI I-D Rekey ACK to the key server through an acknowledgment message.

Key Server and Group Member Communication

When a key server sends the KEK_ACK_REQUESTED attribute in the KEK SA payload, a group member must respond to subsequent rekey messages with the GDOI I-D Rekey ACK unless notified otherwise by the corresponding key server. The communication between a key server and group members are as follows:

1. For every GROUPKEY-PUSH message sent by a key server, the group member must respond with the GROUP-PUSH-KEY ACK message.
2. The key server verifies and validates the message for format and payload. If validation fails, the message is dropped.
3. If validation is successful, the key server processes the SEQ and ID payloads to record the latest acknowledged sequence number for the group member associated with the ID. The sequence number must be the same as the last sent sequence number; otherwise, the SEQ and ID payload will not be recorded.

Note In case of a Cisco key server, a group member is removed from the database if a group member does not send an acknowledgment for three consecutive rekey messages. If a group member is configured with the unicast rekey feature and the KEK_ACK_REQUESTED attribute is not sent for a given KEK Security Parameter Index (SPI), the group members must send the Cisco Unicast Rekey ACK message to the key server.

The following table explains the attributes sent in the KEK SA payload along with the values sent for each acknowledgment option configured on a key server:

Table 25: KEK SA Payload for Each Acknowledgment Option

Acknowledgement Option	New Cisco Group Member	Cisco ASR 9000 Group Member	Non-Cisco Group Member
Cisco	No Attribute	No Attribute	No Attribute
Interoperable	K E K _ A C K _ R E Q REKEY_ACK_KEK_SHA256	KEK_ACK_REQ REKEY_ACK_KEK_SHA256	KEK_ACK_REQ REKEY_ACK_KEK_SHA256
Any	No Attribute	KEK_ACK_REQ REKEY_ACK_KEK_SHA256	KEK_ACK_REQ REKEY_ACK_KEK_SHA256

Note When the **no rekey acknowledgement** command is used to set the rekey acknowledgment to the default value ‘Cisco’, the key server does not include the KEK_ACK_REQUESTED attribute in the KEK SA payload.

The following table explains the acknowledgment methodology for each acknowledgment type configured via the keywords in the **rekey acknowledgement** command on a key server:

Table 26: Acknowledgment Methodology

Acknowledgement Option	Key Server Accepts I-D ACK	Key Server Accepts Cisco ACK
Cisco	No (results in error)	Yes
Interoperable	Yes	No (results in error)
Any	Yes	Yes

How to Configure GET VPN Interoperability

Ensuring the Correct GDOI Version on a Key Server

SUMMARY STEPS

1. **enable**
2. **show crypto gkm feature** *feature name*
3. **show crypto gkm feature** *feature-name* | **include no**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

```
Device> enable
```

Step 2 **show crypto gkm feature** *feature name*

Displays the GDOI version running on each key server and group member in the network and information about whether the device supports GET VPN interoperability features, namely, D3P support on GETVPN Key Server and Internet-Draft ACK for Cisco GETVPN Key Server.

Example:

```
Device# show crypto gkm feature ip-d3p
Group Name: GET VPN1
    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
```

Example:

```
Device# show crypto gkm feature gdoi-interop-ack
Group Name: GET VPN2
    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
```

Step 3 **show crypto gkm feature** *feature-name* | **include no**

(Optional) Finds devices that do not support a feature.

Example:

```
Device# show crypto gkm feature gdoi-interop-ack | include no
```

Ensuring the Correct GDOI Version on a Group Member

SUMMARY STEPS

1. **enable**
2. **show crypto gkm feature** *feature name*

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show crypto gkm feature** *feature name*

Displays the GDOI version running on a group member in the network and information about whether the device supports GET VPN interoperability features, namely, D3P support on GETVPN Key Server and Internet-Draft ACK for Cisco GETVPN Key Server.

Example:

```
Device# show crypto gkm feature ip-d3p
       Version    Feature Supported
        1.0.11           Yes
```

Example:

```
Device# show crypto gkm feature gdoi-interop-ack
       Version    Feature Supported
        1.0.10           No
```

Enabling IP-D3P on a Key Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group GETVPN**
4. **server local**
5. **sa d3p window**{**sec** *seconds* | **msec** *milliseconds*}
6. **exit**
7. **show crypto gkm ks replay**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gkm group GETVPN** **Example:** `Device(config)# crypto gkm group GETVPN`	Configures a group key management (GKM) group and enters GKM group configuration mode.
Step 4	**server local** **Example:** `Device(config-gkm-group)# server local`	Designates the device as a key server and enters GDOI local server configuration mode.
Step 5	**sa d3p window** {**sec** *seconds* | **msec** *milliseconds*} **Example:** `Device(gdoi-local-server)# sa d3p window msec 5000`	Enables IP delivery delay detection protocol (IP-D3P) on all security associations in the group. • **sec** *seconds*—Window size, in seconds. The range is from1 to 100. • **msec** *milliseconds*—Window size, in milliseconds. The range is from 100 to 10000.
Step 6	**exit** **Example:** `Device(gdoi-local-server)# exit`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.
Step 7	**show crypto gkm ks replay** **Example:** `Device# show crypto gkm ks replay`	Displays key server group information for time-based anti-replay.

Example

The following is a sample output from the **show crypto gkm ks replay** command:

```
Device# show crypto gkm ks replay
Anti-replay Information For Group GETVPN:
    IP-D3P: Type = POSIX-TIME-MSEC, Window-size = 5000 msec
```

Enabling IP-D3P on a Group Member

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group GET**
4. **client d3p window** {**sec** *seconds* | **msec** *milliseconds*}
5. **exit**
6. **show crypto gkm gm replay**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gkm group GET** **Example:** `Device(config)# crypto gkm group GETVPN`	Configures a group key management (GKM) group and enters GKM group configuration mode.
Step 4	**client d3p window** {**sec** *seconds* | **msec** *milliseconds*} **Example:** `Device(config-gkm-group)# client d3p window sec 50`	Enables client-acceptable IP delivery delay detection protocol (IP-D3P). • **sec** *seconds*—Window size, in seconds. The range is from1 to 100. • **msec** *milliseconds*—Window size, in milliseconds. The range is from 100 to 10000.
Step 5	**exit** **Example:** `Device(gdoi-local-server)# exit`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.
Step 6	**show crypto gkm gm replay** **Example:** `Device# show crypto gkm gm replay`	Displays group member information for time-based anti-replay.

Example

The following is a sample output from the **show crypto gkm gm replay** command:

```
Device# show crypto gkm gm replay
Anti-replay Information For Group GET:
   IP-D3P:
     Posix-time-msec           : 502764.17
     Input Packets             : 5        Output Packets           : 5
     Input Error Packets       : 5        Output Error Packets     : 0

IP-D3P Error History (sampled at 10pak/min):
     xx:xx:xx.xxx PST Tue Feb 25 2014: src=5.0.0.2; my_time=502729.95; peer_time=33.46;
win=10
     yy:yy:yy.yyy PST Tue Feb 25 2014: src=5.0.0.2; my_time=502723.95; peer_time=27.45;
win=10
```

Enabling Rekey Acknowledgment

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto gkm group GET**
4. **server local**
5. **rekey acknowledgement**{**cisco** | **interoperable** | **any**}
6. **exit**
7. **show crypto gkm ks replay**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto gkm group GET** **Example:** `Device(config)# crypto gkm group GET`	Configures a group key management (GKM) group and enters GKM group configuration mode.
Step 4	**server local** **Example:** `Device(config-gkm-group)# server local`	Designates the device as a key server and enters GDOI local server configuration mode.
Step 5	**rekey acknowledgement**{**cisco** | **interoperable** | **any**} **Example:** `Device(gdoi-local-server)# rekey acknowledgment interoperable`	Enables group members to acknowledge rekeys. • **cisco**—Accepts Cisco Rekey ACK (encrypted) message.

	Command or Action	Purpose
		• **interoperable**—Requests and accepts interoperable rekey ACK (unencrypted) message. • **any**—Accepts a supported ACK message based on group key member version.
Step 6	**exit** **Example:** `Device(gdoi-local-server)# exit`	Exits GDOI local server configuration mode and returns to privileged EXEC mode.
Step 7	**show crypto gkm ks replay** **Example:** `Device# show crypto gkm ks replay`	Displays rekey acknowledgment configuration on the key server.

Example

The following is a sample output from **show** commands displaying the rekey acknowledgment configuration:

```
Device# show crypto gkm

GROUP INFORMATION
    Group Name               : GETVPN (Unicast)
 .
 .
 .
    Group Rekey Lifetime     : 86400 secs
    Group Rekey
        Remaining Lifetime   : 44710 secs
        Time to Rekey        : 44485 secs
        Acknowledgement Cfg  : {Cisco|Interoperable|Any}
 .
 .
 .

Device# show crypto gkm ks

Total group members registered to this box: 0
Key Server Information For Group GETVPN:
    Group Name               : GETVPN
Group Name               : GETVPN (Unicast)
 .
 .
 .
    Group Members            : 0
      GDOI Group Members     : 0
      G-IKEv2 Group Members  : 0
    Rekey Acknowledgement Cfg: {Cisco|Interoperable|Any}
    IPSec SA Direction       : Both
 .
 .
 .
Device# show crypto gkm ks rekey

Group GETVPN (Unicast)
```

```
    Acknowledgement Type In-Use         : {Cisco|Interoperable|Any}
    Number of Rekeys sent               : 20
 .
 .
 .
Device# show crypto gkm ks rekey

Group GETVPN (Multicast)
    Acknowledgement Type In-Use         : None
    Number of Rekeys sent               : 20
 .
 .
 .
Device# show crypto gkm ks policy

Key Server Policy:
For group diffint (handle: 2147483650) server 10.0.8.1 (handle: 2147483650):
  # of teks : 2  Seq num : 7
  KEK POLICY (transport type : Unicast)
    spi : 0x7D32D2052B87CEFE14060B58B0176129
    management alg     : disabled    encrypt alg       : AES
    crypto iv length   : 16          key size          : 16
    orig life(sec): 86400       remaining life(sec): 44699
    time to rekey (sec): 44474
    sig hash algorithm : enabled     sig key length    : 162
    sig size           : 128
    sig key name       : mykeys
    acknowledgement    : {cisco|interoperable|any}

Device# show crypto gkm ks policy

Key Server Policy:
For group diffint (handle: 2147483650) server 10.0.8.1 (handle: 2147483650):
  # of teks : 2  Seq num : 7
  KEK POLICY (transport type : Multicast)
    spi : 0x7D32D2052B87CEFE14060B58B0176129
    management alg     : disabled    encrypt alg       : AES
    crypto iv length   : 16          key size          : 16
    orig life(sec): 86400       remaining life(sec): 44699
    time to rekey (sec): 44474
    sig hash algorithm : enabled     sig key length    : 162
    sig size           : 128
    sig key name       : mykeys
    acknowledgement    : none
```

Configuration Examples for GET VPN Interoperability

Example: Enabling IP-D3P on a Key Server

```
Device> enable
Device# configure terminal
Device(config)# crypto gkm group GETVPN
Device(config-gkm-group)# server local
Device(gdoi-local-server)# sa d3p window msec 5000
Device(gdoi-local-server)# exit
```

Example: Enabling IP-D3P on a Group Member

```
Device> enable
Device# configure terminal
Device(config-gkm-group)# client d3p window sec 50
Device(gdoi-local-server)# exit
```

Example: Enabling Rekey Acknowledgement

```
Device> enable
Device# configure terminal
Device(config)# crypto gkm group GET
Device(config-gkm-group)# server local
Device(gdoi-local-server)# rekey acknowledgment interoperable
Device(gdoi-local-server)# exit
```

Additional References for GET VPN Interoperability

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
GET VPN configuration	*Cisco Group Encrypted Transport VPN*
Unicast rekey	"Unicast Rekeying" section in the *GET VPN* module

Standards and RFCs

Standard/RFC	Title
draft-weis-delay-detection-00	*IP Delivery Delay Detection Protocol*
draft-weis-gdoi-rekey-ack-01	*GDOI GROUPKEY-PUSH Acknowledgement Message*
RFC 5374- Section 5.4 - Group Associated Policy	*Multicast Extensions to the Security Architecture for the Internet Protocol*
RFC 6407 - Section 4.2.1 - Activation Time Delay	*The Group Domain of Interpretation*

Technical Assistance

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

Feature Information for GET VPN Interoperability

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

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

Table 27: Feature Information for GET VPN Interoperability

Feature Name	Releases	Feature Information
D3P support on GETVPN Key Server	Cisco IOS XE Fuji 16.7.1	The D3P support on GETVPN Key Server feature enables support for IP-D3P on a GET VPN network. The following commands were introduced or modified: **client d3p**, **sa d3p**, **show crypto gkm gm replay**, **show crypto gkm ks replay**.
Internet-Draft ACK for Cisco GETVPN Key Server	Cisco IOS XE Fuji 16.7.1	The Internet-Draft ACK for Cisco GETVPN Key Server implements the standard for rekey acknowledgment message between non-Cisco group members and key server as defined in the GDOI GROUPKEY-PUSH Acknowledgment Message draft. The following commands were introduced or modified: **rekey acknowledgement**, **show crypto gkm**.
RFC 8263 ID Ack implementation	Cisco IOS XE Amsterdam 17.1.1	The Group Domain of Interpretation (GDOI) includes the ability of key server to provide a set of current devices with additional security associations. For example, to rekey expiring security associations. This feature adds the ability of a key server to request that the group devices return an acknowledgement of receipt of its rekey message and specifies the acknowledgement method.