



Expertise
and insight
for the future

Arun Katuwal

Deploying and Testing IKEv2, Flex VPN and GET VPN

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

18 November 2017

Author(s) Title	Arun Katuwal Deploying and Testing IKEv2, Flex VPN and GET VPN
Number of Pages Date	36 pages + 6 appendices 18 November 2017
Degree	Bachelor's Degree
Degree Programme	Information Technology
Specialisation option	Security and Networking
Instructor(s)	Erik Pätynen, Senior Lecturer
<p>The goal of this thesis was to study the detailed configuration and deployment methods of IKEv2, FlexVPN and GET VPN so that any reader can test it in a real environment and evaluate and analyze the comparison between these technologies. This thesis is aimed not only to study the configuration methods but also to deploy FlexVPN using IKEv2 in a real environment at the laboratory for the testing.</p> <p>Initially, the first phase of this project was to study the detailed configuration and deployment methods of IKEv2. As IKEv2 is the building blocks for Flex VPN, detail information was necessary. To complete this project, a network topology was designed and was implemented in the school laboratory. During this project, Cisco routers were used, and Windows desktop computers were used as the end devices. During the laboratory work site to site Flex VPN configuration were established to get the results. The command line interface was used as a tool to configure and the network connection was successfully established. The established connection was secured from end to end.</p> <p>Due to some limitations, the network that was designed reflects a very small size organization network. But the methods of implementation are applicable to a network of any size. After implementing everything, the network was tested, and it performed as designed, the tunnel which was implemented between the routers were also successful.</p>	
Keywords	IKEv2, FlexVPN, GET VPN, tunnel, deployment, configuration

Contents

1	Introduction	1
2	VPN Overview	2
2.1	VPN Terminology	2
2.2	Different VPN Technologies	3
2.3	Types of VPN Protocols	3
3	IPSec v2 VPN	6
3.1	Introduction to IPSec VPNs	6
3.2	The Internet Key Exchange Protocol Version 2(IKEv2)	6
3.2.1	Benefits of IKEv2	7
3.3	IKEv2 Configuration	8
4	FlexVPN	11
4.1	FlexVPN Building Blocks	11
2.2.1	IKEv2	11
2.2.2	Cisco IOS point to point(P2P) tunnel interfaces	11
2.2.3	Cisco IOS AAA infrastructure	13
4.2	Benefits of FlexVPN	15
4.3	FlexVPN Configuration	16
4.3.1	FlexVPN Server	16
4.3.2	FlexVPN Client	19
5	GET VPN	23
5.1	GET VPN Overview	23
5.2	GET VPN Architecture	24
5.2.1	Key Distribution Group Domain of Interperation	25
5.2.2	Address Preservation	26
5.2.3	Secure Data Plane Multicast	27
5.2.4	Secure Data Plane Unicast	27
6	FlexVPN Deployment	28
6.1	Requirements and Topology	28
6.2	Configurations	29
6.3	Results	33

7	Conclusions and Discussion	35
	References	36

Appendices

Appendix 1. Branch 1 Running Configuration

Appendix 2. Branch 2 Running Configuration

Appendix 3. Internet Running Configuration

Appendix 4. IKEv2 Configurations

Appendix 5. FlexVPN Configurations

Appendix 6. GET VPN Configurations

ABBREVIATIONS/ACRONYMS

AAA	Authentication Authorization Accounting
DOI	Domain of Interpretation
DPD	Dead Peer Detection
EAP	Extensible Authentication Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
FQDN	Fully Qualified Domain Name
FVRF	Transport VRF
GET VPN	Group Encrypted Transport VPN
GRE	Generic Routing Encapsulation
IETF	Internet Engineering Task Force
IKE	Internet Key exchange protocol
IKEv2	Internet Key Exchange Version 2
IPSec	Internet Protocol Security
ISAKMP	Internet security association and key management protocol
IVRF	Overlay VRF
L2F	Layer Two forwarding protocol
L2TP	Layer 2 Tunneling Protocol
MPLS	Multiprotocol Label Switching
OSPF	Open Shortest Path First
PPTP	Point-to-Point Tunneling Protocol
SAs	Security Associations
SSL	Secure Sockets Layers
VPN	Virtual Private Network
VRF	VPN routing and forwarding
WAN	Wide Area Network

1 Introduction

In the current situation, everything around us is getting connected with one another using networks. So, the Information Technology (IT) companies and users all over the globes are concerned about the Network Security. The concept of Virtual private network was established due to the necessity of network security. A Virtual Private Network (VPN) is a network that is created by connecting many sites that are installed on shared network with the similar security as a private network. Different VPN technologies such as IPsec (Internet Protocol Security) VPN, MPLS (Multiprotocol Label Switching) VPN, SSL (Secure Sockets Layers) VPN, FlexVPN and Tunnel-less VPNs are available at present time. A VPN can erase all kinds of location barriers and allows employees to work remotely and efficiently from the place of their residence and allows a business to connect securely with its providers and partners.

The main objective of this thesis project is to study the detailed configuration and deployments methods of Internet Key Exchange Version 2(IKEv2), FlexVPN and Group Encrypted Transport VPN (GET VPN) so that any reader can test it in real environment and evaluate and analyze the comparison between these technologies. Moreover, IKEv2 and FlexVPN is deployed in real environment for the testing.

This thesis consists of six sections. The first and second section is the introduction of the topic and basic overview of VPNs. The third section describe the theoretical background about the IPsec VPNs and IKEv2, it also describes the configuration methods. The fourth section describe about the FlexVPN, FlexVPN Server and FlexVPN Client and the configuration methods. Fifth section is all about the theoretical background and architecture of GET VPN. The sixth section describe about the Deployments of FlexVPN in real environment at the school laboratory. The final section comprises of the conclusion, where the facts relating to this thesis research is summarized.

2 VPN Overview

2.1 VPN Terminology

VPN allows user to securely extend a private network across an untrusted network. It is called as Virtual Private Network because the private network is virtually extended by a logical private connection. A private network which is connected using the public communication infrastructure like internet, maintaining the privacy by following certain security and tunneling protocols is known as Virtual Private Network. VPNs are used by most of the companies for their voice and data communication as it is cheaper alternative of expensive owned networks as it uses the shared public networks.

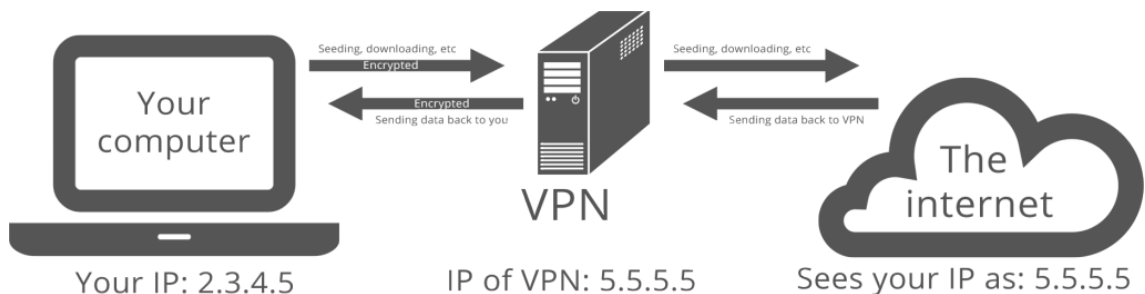


Figure 1: A simple VPN diagram [8]

The traffic inside a tunnel cannot be identified if a VPN is used, traffic can be anything that is sent over the internet. The devices situated in the path of the network can only see the VPN traffic even after the VPN is routed in the internet. The devices that are situated in the path are totally uninformed about the transmitting information in the inner core of the private tunnel. Secured protocols for example HTTPS and SSH are still guarded from other VPN clients inside the tunnel and will also be unidentifiable from outside the tunnel. VPN encrypts, hides and protects the traffic within and individual information from those outside the tunnel. [1]

2.2 Different VPN Technologies

Site-to-Site VPN

A Site-to-Site VPN is mostly used by corporates and organizations, it is often known as Router-to-Router VPN. Different organization have their workplaces in various geological areas and all those organization utilize this VPN to connect the network of one office location to the other location. There are two types, one is Intranet based VPN and another Extranet based VPN. If various branch office of a main office situated at different locations are connected using Site-to-Site VPN then it is called Intranet based VPN. And if different companies connect themselves using Site-to-Site VPN is called Extranet based. Site-to-site VPN makes a virtual bridge between the networks and connects them through internet with a protected and private connection between the systems. In Site-to-Site VPN one of the router behaves as a VPN Client while another behaves as a VPN Server.

Remote Access VPN

Remote access VPN enables a user to connect with a private network and lets the user to utilize its resources and service remotely. The secure connection between the client and the private network occurs through the Internet. This VPN is useful for both Business Company and a home user. VPN can be utilized by a company worker during vacation to get access to the files and the resources on the private network remotely. Users that are aware of internet security also uses VPN service to improve their security and privacy.

2.3 Types of VPN Protocols

Layer 2 Tunneling Protocol (L2TP)

The layer two tunneling protocol depends on the concept of Layer Two forwarding protocol (L2F). It is a tunneling protocol that is mostly and frequently combined with some other VPN security protocol in order to create an extremely secure VPN connection. At transport layer, it encloses an entire data-link-layer frame into a UDP. Hence, a data

packet with a private or local address can be transferred along the internet. The above mentioned UDP packet consists of UDP header with control bits, version and the length of the packet, a sequence number and tunnel-id to monitor the current VPN connection to be sure of correct packet processing. After this process, MAC address and the payload are followed by layer two frame. UDP packet containing a solo encapsulation of a layer two frame does not provide data privacy. Thus, time and again L2TP is combined with the IPsec. [2]

Point-to-Point Tunneling Protocol (PPTP)

PPTP was developed by Microsoft and Ascend in 1999 and it is one of the oldest VPN. This point-to-point tunneling protocol is an expansion to the point-to-point protocol (PPP) and it is supported by all versions of Microsoft Windows. PPTP utilizes two different packets to set up a VPN connection. Initially, VPN are carried out by the Generic Routing encapsulation (GRE) packets and it adds the GRE header to the first packet. The GRE header is like the L2TP header which contains different control bits, tunnel and sequence numbers. PPTP control message is the second packet type. Simply, this is a TCP packet that has control information, for example, connection parameters, and connection request and response and error messages. PPTP must combine with additional security techniques because GRE does not provide authentication and encryption. [2]

Secure Sockets Layers (SSL)

SSL-based VPNs are also called client-less VPNs or internet-based VPNs. It is one of the most commonly utilized VPNs these days even though there are a few vendors that provide separate client software and these depends on the SSL/TLS protocol. Most SSL-based VPNs utilize a similar set of rules as is used for secure site (HTTPS), whereas Open VPN uses a certain format for encryption. There is no well categorized standard for SSL-based VPNs, so the SSL/TLS protocol is used to set up the secure connection. The connection is secured with one-time password or username. SSL-based VPNs are fundamentally the same as the connection used to secure sites and a similar protocol is frequently used. [2]

Internet Protocol Security (IPSec)

The internet protocol packets are not secure as the data inside it can be altered during the process of communication. IPSec gives a compilation of standard set of rules and methods to set up secure VPNs and provides transmission security at the IP layer. Transport mode and tunnel mode are the two main and essential modes of IPSec connection. The authentication and integrity information in the IPSec header is added to the original IP header when IPSec is operated using the transport mode. While the tunnel mode maintains more flexibility as every unique IP protocol is enclosed by another IP packet that contains of a new IPSec header and the new IP header.

IPSec connections and exchange encryption keys are built using the internet key exchange protocol (IKE) and it is also used to share the authentication data. Both tunnel partners arrange the parameters of the VPN association so that they can use a common SA. IKE messages are exchanged by means of UDP bundles at port 500 and depend on the Internet security association and key management protocol (ISAKMP). [2]

3 Internet Protocol Security (IPSec) Version 2

3.1 Introduction to IPSec VPNs

A set of protocols that are utilized to establish a secure internet connection, transmission of data and communications, or basically, internet is known as Internet Protocol Security (IPSec). IPSec works by validating and encoding each bundle of information to set up a secure connection.

For the sake of connectivity between site locations or for remote mobile workers, IPsec VPNs were commonly employed. It requires secure connectivity from hotels, where special software was required. Due to advancement in technology and internet, VPN of communication has rapidly increased and is expected to rise in a long run time. Such increment is because of the ability to connect, communicate with and remotely manage every IP-enabled device over a secure medium.

When IPSec VPNs are used, the traffic will be protected to ensure that an observer cannot view the plaintext data, this is achieved by encryption that provides confidentiality. IPsec can use Authentication Header or Encapsulation Security Payload to provide security services to protect the IPsec Security Association. IPsec Tunnel mode allows for traffic to be protected within the IPsec Security Association, which performs limited traffic flow confidentiality by hiding the internal IP address, allowing for traffic to be tunneled within the IPsec Security Association. However, it requires an additional IP header. Transport mode requires that the protected traffic's IP header is used to transport the packet on the wire, but it results in less overhead than Tunnel mode. [3]

3.2 The Internet Key Exchange Protocol Version 2(IKEv2)

Internet Key Exchange Version 2 is the second-generation standard for a secure key exchange between connected devices. To build up a secure connection, IKEv2 utilizes an IPSec-based tunneling protocol. During the disturbance in the VPN connection IKEv2 can re-establish the connection with a rapid speed, it is the power and the most useful advantage of using IKEv2. IKEv2 makes possibility to use on Windows or iOS because of the features like fast re-connection and strong encryption.

Currently, IKEv2 is one of the best protocol compared to other VPN protocols for mobile clients. At the point when it is utilized together with Microsoft's VPN Connect, the IKEv2 protocol will automatically re-build up the VPN connection if the established connection is disconnected. This implies that when the internet is suddenly disturbed, or the connection fails because of some reasons then the VPN connection will establish the internet connection is split of a second. The IKEv2 VPN convention will work brilliantly even if the user switch between systems as it supports Mobility and Multihoming protocol. This is especially important for those users that like to utilize their cell phones frequently and frequently move from their mobile network with a Wi-Fi connection or even utilize hotspots.

IKEv2 protocol is less in use now than IPSec because it is supported on fewer platforms. On the other hand, the IKEv2 is better than other great protocols because of security, stability, speed and the power to re-establish a connection. Comparatively, IKEv2 VPN is less popular than another existing VPNs because this is new protocol and is not supported in as many devices as other existing VPNs. Yet, the popularity is expanding because of things it is prepared to do. And increasing number of clients are starting to utilize it as it continuously develops and will most likely be utilized as a part of more gadgets in the near future. [4]

3.2.1 Benefits of IKEv2

- Internet Key Exchange Version 2 (IKEv2) produce an in-built support for Dead Peer Detection (DPD) and Network Address Translation-Traversal (NAT-T).
- Certificate URLs
To avoid demoralization, URL can be used to reference the Certificates, instead of being sent within IKEv2 packets.

- Denial of Service Attack Resilience
Until and unless IKEv2 determines the requester, IKEv2 does not process a request, 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.
- EAP Support
The use of Extensible Authentication Protocol (EAP) for authentication is supported by IKEv2.
- 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): To provide reliability and directs some error-processing logistics, IKEv2 conducts sequence numbers and acknowledgments to share state management.

3.3 IKEv2 Configuration

The main goal of the Cisco IKEv2 configuration is to keep the configuration intuitive and minimal. The user needs to configure only the mandatory parameters, while the smart defaults are used wherever possible.

- Configuring Global IKEv2 Options

To configure global IKEv2 options enter the global configuration mode and use the command `crypto ikev2 {add rest here}` to configure the certificate number, the url, cookie challenge number and so on.

- Configuring the IKEv2 Proposal

Follow the below mentioned process to configure the proposals, there are two options. Option one being using the default proposal and option two is mentioned below. The default IKEv2 proposal requires no configuration and is a collection of commonly used transforms types, which are as follows:

```
Encryption                aes-cbc-128                3des
integrity sha md5
group 5 2
```

For manual configuration, enter the global configuration mode and give a proposal name. There are sets of encryption and integrity and group number, the number can be chosen accordingly.

- Configuring the IKEv2 Policy

After you create the IKEv2 proposal, the proposal must be attached to a policy to pick the proposal for negotiation. The default proposal associated with the default policy is used for negotiation. An IKEv2 policy with no proposal is considered incomplete. During the initial exchange, the local address (IPv4 or IPv6) and the FVRF of the negotiating SA is matched with the policy and the proposal is selected. [4]

- Configuring the IKEv2 Keyring

IKEv2 keyring keys must be configured in the peer configuration sub-mode that defines a peer sub-block. An IKEv2 keyring can have multiple peer sub-blocks. A peer sub-block contains a single symmetric or asymmetric key pair for a peer or peer group identified by any combination of hostname, identity, and IP address. [4]

To configure the IKEv2 keyring use command `crypto ikev2 keyring` and give keyring name add a description name, assign the IP address and give a unique identity using FQDN (Fully Qualified Domain Name) name.

- Configuring the IKEv2 Profile

To configure IKEv2 Profile follow the following steps. After creating proposal, policy and keyring, profile matching must be done. An IKEv2 profile should be attached to an IP-Sec profile or a crypto map on both the IKEv2 initiator and responder. Use the command `set ikev2-profile profile-name` to attach the profile. Then use `fqdn`, `email`, `ip address` or `key name` to match the identity.

Please note that in IKEv2, NAT-T is auto detected. To disable NAT-T encapsulation, use the

`no crypto ipsec nat-transparency udp-encapsulation` command.

Use the `show crypto ikev2 profile tag` command to display the IKEv2 profile

- **Configuring the IKEv2 Name Mangler**

IKEv2 name mangler is used to derive a name for the authorization requests. The name is derived from specified portions of different forms of remote IKE identities or the EAP identity. The name mangler specified here is referred to in the IKEv2 profile. [4]

To configure Name Mangler, enter the global configuration mode and define a name mangler. Then derive the name from any of the fields. Accordingly, derive the name from the remote identity of type eap, email and fdqn.

- **Configuring the IKEv2 Authorization Policy**

The IKEv2 authorization policy serves as a container of IKEv2 local AAA group authorization parameters. The IKEv2 authorization policy is referred from IKEv2 profile via the `aaa authorization group` command. [4]

To configure Authorization Policy, enter the global configuration mode and enter the IKEv2 authorization policy configuration mode. Then specify the dhcp server to lease an IP address, and specify the dns servers and netmasks and pool name. After this process define the ACL for split tunneling and specify the internal Windows Internet Naming Service.

- **Configuring the IKEv2 Fragmentation**

We can use `crypto ikev2 fragmentation [mtu mtu-size]` command to fragment the IKEv2 packets at IKEv2 layer and to avoid fragmentation after encryption. IKEv2 peers negotiate the support for fragmentation and the MTU in the IKE_INIT exchange. Fragmentation of packets exceeding the negotiated MTU starts with IKE_AUTH exchange. [4]

4 FlexVPN

FlexVPN is a configuration framework (a collection of CLI/API commands) aimed to simplify setup of remote access, site-to-site and DMVPN topologies. It is a powerful and standards-based VPN technology that enables giant businesses to securely establish the connection between different offices and remote clients. It delivers outstanding cost saving service compared to the cost of numerous separate types of virtual private network (VPN) solutions such as GRE (generic routing encapsulation), crypto map, and virtual tunnel interface (VTI)-based solutions. FlexVPN relies on open-standards-based (ASR 1000) running IOS-XE, including Cloud Services Router 1000v (CSR 1000v) and ISR4000 series routers. Cisco AnyConnect is supported as a client, in addition to any compatible third-party client. Most of the configuration commands begin with `crypto ikev2` and come with smart defaults, representing Cisco's view of best practice design. Dynamic tunnel configuration has been simplified so that, theoretically, you'd only need a single interface template on the Hub site to allow all types of incoming VPN connections. [3]

4.1 FlexVPN Building Blocks

4.1.1 IKEv2

Flex VPN is based on IKEv2 and hence bring all the IKEv2 protocol features such as configuration exchange, IKEv2 redirect for server load balancing, cookie challenges for DoS mitigation, NAT traversal, IKEv2 fragmentation, and Cisco IOS IKEv2 features, such as IKEv2 call admission control, session deletion on certificate expiry, or revocation to all the supported VPN topologies. FlexVPN makes use of the IKEv2 configuration to exchange policy parameters between peers (typically between FlexVPN client and server) and to exchange routing information between peers. It adds routes to remote overlay addresses and protected subnets and serves as a lightweight overlay routing mechanism. [3]

4.1.2 Cisco IOS Point-to-Point (P2P) Tunnel Interfaces

FlexVPN uses per-peer point-to-point (P2P) tunnel interfaces for all the supported VPN topologies with GRE or native IPsec encapsulation. GRE encapsulation offers the benefit of native support for IP dual stack because it can carry both Ipv4 and Ipv6 overlay traffic over either Ipv4 or Ipv6 transport, whereas native IPsec encapsulation can carry either Ipv4 or Ipv6 overlay traffic over either Ipv4 or Ipv6 transport and is useful for interoperating with implementations that do not support GRE. The P2P tunnel interface is statically configured on FlexVPN initiators and is dynamically instantiated from a virtual-template interface on FlexVPN responders.

Table 1. P2P Tunnel Interface Used by FlexVPN [3]

P2P Tunnel Interface Type	Usage
Static GRE interface (tunnel mode gre)	Initiator
Static Ipsec VTI(sVTI) (tunnel mode Ipsec)	Initiator
Virtual-template interface of type tunnel with GRE encapsulation (tunnel mode Ipsec) Native Ipsec encapsulation (tunnel mode Ipsec) Auto-detection of tunnel mode and transport protocol (Ipv4/Ipv6)	Responder
Virtual-access interface cloned from virtual-template	Responder

- Configuring Static P2P Tunnel Interfaces

The IPsec Security Associations (SAs) originate from an IPsec interface. With FlexVPN, the IPsec interface is a tunnel interface with the tunnel protection applied. The IPsec parameters used in the SA negotiation are derived from the tunnel interface configuration and mainly the tunnel encapsulation mode determines the IPsec Security Association traffic selectors.

The following are the P2P tunnel interface commands on an initiator that determine the outbound IPsec Security Association negotiation parameters.

- The tunnel mode command determines the encapsulation protocol and the transport IP address family (IPv4/IPv6). The address specified in the tunnel source and tunnel destination commands must match this IP address family.

- The tunnel source address is typically the address of the WAN interface. When there are multiple WAN interfaces, it can be a loopback interface IP address that can be reached through all the WAN transports.
- The IP address and ipv6 address commands determine the overlay IP address family (IPv4 or IPv6) i.e. whether the interface can support IPv4 and/or IPv6 overlay traffic.
- The vrf forwarding command specifies the IVRF (overlay VRF). The tunnel vrf command specifies the FVRF (transport VRF).
- The tunnel protection command enables IPsec on the interface and specifies the IPsec protection profile

4.1.3 Cisco IOS AAA Infrastructure

Cisco IOS AAA infrastructure provides an abstracted framework for authentication, authorization, and accounting using the local database or external AAA servers that hold the authentication credentials, authorization policy parameters and accounting data in a way that is transparent to the AAA clients. FlexVPN registers as an AAA client and leverages AAA for EAP authentication, AAA-based pre-shared keys, user and group authorization policy, and accounting.

Table 2. FlexVPN AAA Operations and Supported AAA Database. [3]

FlexVPN aaa Operation	Supported aaa Database
EAP authentication	External aaa server for standards-based EAP External and local aaa for any connect-EAP
aa-based pre-shared keys	External aaa server
User authorization	External aaa server and Local aaa database
Group authorization	External aaa server and Local aaa database
Implicit authorization	External aaa server
Accounting	External aaa server

The Steps to configure AAA for FlexVPN are mentioned below.

- Step 1. Enable AAA, using the aaa new-model command.
- Step 2. Define AAA method list that specifies an ordered list of AAA databases (local AAA database and external AAA servers) to be used for authentication, authorization, and accounting. Note that the next database in the ordered list is

tried only if the previous database is not reachable and not if it returns a failure.

- Step 3. If the method list specifies external AAA server groups, define the server groups and the servers.
- Step 4. If the method list specifies local AAA database, configure the local AAA database for FlexVPN using the `crypto ikev2 authorization policy` command.
- Step 5. Reference the AAA method list from the IKEv2 profile.
- Configuring AAA for FlexVPN
- Here is an example of configuration of aaa for FlexVPN.

1. Enable aaa

```
aaa new model
```

2. Define the aaa method list for authentication, authorization and accounting as needed.

```
aaa authentication login authen_list_radius group radius_group
```

```
aaa authorization network author_list_local local
```

```
aaa authorization network author_list_radius group radius_group
```

```
aaa accounting network acc_list_radius group radius_group
```

3. Define the RADIUS server group

```
aaa group server radius radius_group
```

```
server name radius_server
```

4. Define the RADIUS Server

```
radius server radius_server address ipv4 172.16.1.3
```

```
auth-port 1645 acct-port 1646 key radius_key
```

5. Configure Local aaa database for FlexVPN

```
crypto ikev2 authorization policy author_policy
```

```
route set interface
```

```
route accept any
```

```
pool ip_address_pool
```

```
aaa attribute list attr_list
```

6. Define the aaa attribute list

```
aaa attribute list attr-list1
attribute type interface-config "ip mtu 1100"
attribute type interface-config "tunnel key 10"
```

7. Configure FlexVPN authentication, authorization and accounting in the IKEv2 profile as needed.

```
crypto ikev2 profile ikev2_profile
aaa authentication eap authen_list_radius
aaa authorization user cert list author_list_radius
user1
aaa authorization group cert list author_list_local
author_policy
aaa accounting cert acc_list_radius
```

4.2 Benefits of FlexVPN

Flex VPN has various benefits. Some of them are mentioned below:

- FlexVPN can be deployed on both public network, which is also called as Internet and also on a private network like Multiprotocol Label Switching (MPLS) VPN network.
- This VPN is intended for site-to-site and remote access VPNs, a single deployed FlexVPN can acknowledge the two kinds of connection requests at the same time.
- FlexVPN can be implemented with different kinds of redundancy models.
 - Dynamic routing protocols such as OSPF, EIGRP and BGP can be used while implementing FlexVPN tunnels.
- The IT world is changing towards cloud based as well as mobile-based computing. So different types of providers are creating and developing VPN routers and VPN devices. The Cisco IOS FlexVPN solution is compatible with VPN clients from Android devices and Apple iOS and any IKEv2-based third-party VPN providers.

- By default, FlexVPN supports IP Multicast in two ways:
 - IP Multicast packets for each spoke is replicated by the FlexVPN hub router.
- Multicast packet replication is done by transport network after performing IPsec encryption, if the transport network supports native IP Multicast.
- QoS (Quality of Service) can be integrated at per tunnel or per SA basis by the architecture of Cisco IOS FlexVPN.
- FlexVPN is based on IKEv2, this results in improved performance. It is easy to configure by using IKEv2 smart defaults, it has a built-in default. [6]

4.3 FlexVPN Configuration

4.3.1 FlexVPN Server

The FlexVPN server acts as a VPN headend for the remote-access and hub-spoke VPN topologies. FlexVPN server supports peer authentication using the Extensible Authentication protocol (EAP) and acts as a pass-through authenticator relaying EAP messages between the client and the backend EAP server. The backend EAP server is typically a RADIUS server that supports EAP authentication. The FlexVPN server is configured to authenticate FlexVPN clients that use EAP by configuring the authentication remote eap command in IKEv2 profile configuration mode. FlexVPN clients authenticate using EAP by skipping the AUTH payload in the IKE_AUTH request.

If the query-identity keyword is configured, the FlexVPN server queries the EAP identity from the client; otherwise, the FlexVPN client's IKEv2 identity is used as the EAP identity. However, if the query-identity keyword is not configured and the FlexVPN client's IKEv2 identity is an IPv4 or IPv6 address, the session is terminated because IP addresses cannot be used as the EAP identity.

The FlexVPN server starts the EAP authentication by passing the FlexVPN client's EAP identity to the EAP server; the FlexVPN server then relays EAP messages between the remote access (RA) client and the EAP server until the authentication is complete. If the authentication succeeds, the EAP server is expected to return the authenticated EAP identity to the FlexVPN server in the EAP success message. [5]

After EAP authentication, the EAP identity used for the IKEv2 configuration is obtained from the following sources in the given order:

- The EAP identity provided by the EAP server with the EAP success message.
- The EAP identity queried from the client when the query-identity keyword is configured.
- The FlexVPN client IKEv2 identity used as the EAP identity.

IKEv2 RA client	IKEv2 RA server	RADIUS-EAP server
HDR, SAi1, KEi, Ni →		
	← HDR, SAr1, KEr, Nr, [CERTREQ]	
HDR, SK {IDi, [CERTREQ,] [IDr,] SAi2, TSi, TSr} →		
	RADIUS Access-Request/EAP-Message/EAP-Response/(EAP-ID(IKEv2-ID)) →	
		← RADIUS Access-Challenge/EAP-Message/EAP-Request/(EAP-method)
	← HDR, SK {IDr, [CERT,] AUTH, EAP(EAP-Request(EAP-method))}	
HDR, SK {EAP(EAP-Response(EAP-method))} →		
	RADIUS Access-Request/EAP-Message/EAP-Response/EAP-method) →	
		← RADIUS Access-Accept/EAP-Message/EAP-Success (other attributes)
	← HDR, SK {EAP (success)}	
HDR, SK {AUTH} →		
	← HDR, SK {AUTH, SAr2, TSi, TSr }	

200940

Figure 2. IKEv2 Exchange without the query-identity keyword. [5]

IKEv2 RA client	IKEv2 RA server	RADIUS-EAP server
HDR, SAi1, KEi, Ni →		
	← HDR, SAr1, KEr, Nr, [CERTREQ]	
HDR, SK {IDi, [CERTREQ,] [IDr,] SAi2, TSi, TSr} →		
	← HDR, SK {IDr, [CERT,] AUTH, EAP (EAP-request (Identity)) }	
HDR, SK {EAP(EAP-Response(Identity))} →		
	RADIUS Access-Request/ EAP-Message/EAP-Response (EAP-ID) →	
		← RADIUS Access-Challenge/EAP-Message/ EAP-Request/(EAP-method)
	← HDR, SK {IDr, [CERT,] AUTH, EAP(EAP-Request(EAP-method))}	
HDR, SK {EAP(EAP-Response(EAP-method))} →		
	RADIUS Access-Request/EAP-Message/EAP-Response/EAP-method) →	
		← RADIUS Access-Accept/EAP-Message/EAP-Success (EAP-identity) (other attributes)
	← HDR, SK {EAP (success)}	
HDR, SK {AUTH} →		
	← HDR, SK {AUTH, SAr2, TSi, TSr }	

20044

Figure 3. IKEv2 Exchange with the query-identity Keyword. [5]

IKEv2 configuration mode allows IKE peers to exchange configuration information such as IP addresses and routes. The configuration information is obtained from IKEv2 authorization. Both pull, and push models are supported. The pull model involves the exchange of configuration requests and replies; the push model involves the exchange of configuration sets and acknowledgements.

IKEv2 authorization provides a policy for an authenticated session by using the AAA. The policy can be defined locally or on the RADIUS server and contains local and/or remote attributes. The username for authorization can either be derived from the peer identity using the name-mangler keyword or be directly specified in the command. IKEv2 authorization is mandatory only if the peer requests an IP address via configuration mode. An IKEv2 authorization policy defines the local authorization policy and contains local and/or remote attributes. Local attributes, such as VPN routing and forwarding (VRF) and the QOS policy, are applied locally. Remote attributes, such as routes, are pushed to the peer via the configuration mode. Use the `crypto ikev2 author-`

`ization policy` command to define the local policy. The IKEv2 authorization policy is referred from the IKEv2 profile via the `aaa authorization` command.

The IKEv2 name mangler is used to derive the username for IKEv2 authorization and obtain the AAA preshared key from the peer IKE identity.

The FlexVPN server interoperates with the Microsoft Windows7 IKEv2 client, Cisco IKEv2 AnyConnect client, and Cisco FlexVPN client.

- **Configuring the IKEv2 Profile for the FlexVPN Server**

IKEv2 profile commands required for configuring the FlexVPN server in addition to the basic IKEv2 profile commands. To configure the Profile at first enter the IKEv2 profile configuration mode and then specify `aaa authentication` for eap authentication, specify the local or remote authentication followed by specifying the `aaa method list` and username for user and group authorization and enable configuration exchange option.

- **Configuring the IKEv2 Name Mangler**

Name Mangler is used to derive a name for authorization requests and obtain AAA preshared keys. The name is derived from specified portions of different forms of remote IKE identities or the EAP identity. The name mangler specified here is referred to in the IKEv2 profile. At first you need to define a name mangler and enter to name mangler configuration mode then derive the name from any fields in the remote identity of type DN (Distinguished Name). Then derive the name from the remote identity of type EAP, email and fqdn respectively.

- **Configuring the IKEv2 Authorization Policy**

To configure the Authorization Policy at first enter the IKEv2 authorization policy configuration mode and specify `aaa attribute list`, banner, url, default domain, dhcp server and ip address of primary and secondary dns server. Then include a local LAN and specify the maximum number of IPsec SAs, netmask and route set parameters.

4.3.2 FlexVPN Client

The FlexVPN client is a Cisco IOS router-based VPN client positioned for remote offices and mobile workers to provide VPN connectivity to the corporate headquarters. FlexVPN client uses a point-to-point tunnel interface with native IPsec or GRE encapsulation to connect to the FlexVPN server and supports EAP as a local authentication method along with pre-shared key and certificate-based authentication methods. It is a hardware-based VPN client compared to other VPN clients such as Cisco AnyConnect and Microsoft Windows Ikev2 client that are software based. A software VPN client provides VPN access to a single host but the hardware-based FlexVPN client acts as a VPN gateway and provides VPN access for multiple hosts behind the client. [3]

The IKEv2 FlexVPN Client feature establishes a secure IPsec VPN tunnel between a FlexVPN client and a FlexVPN server. The IKEv2 FlexVPN Client feature provides the following benefits:

- Unified tunnel infrastructure
- IPv4/IPv6 proxy support over IPv4/IPv6 transport
- Backward compatibility with some features supported by EasyVPN
- Flexibility for running dynamic routing protocols

Each FlexVPN client is associated with a unique tunnel interface, which implies that the IPsec security association (SA) retrieved by the specific FlexVPN client is bound to the tunnel interface. The figure below shows the association between the FlexVPN client and the tunnel interface. [4]

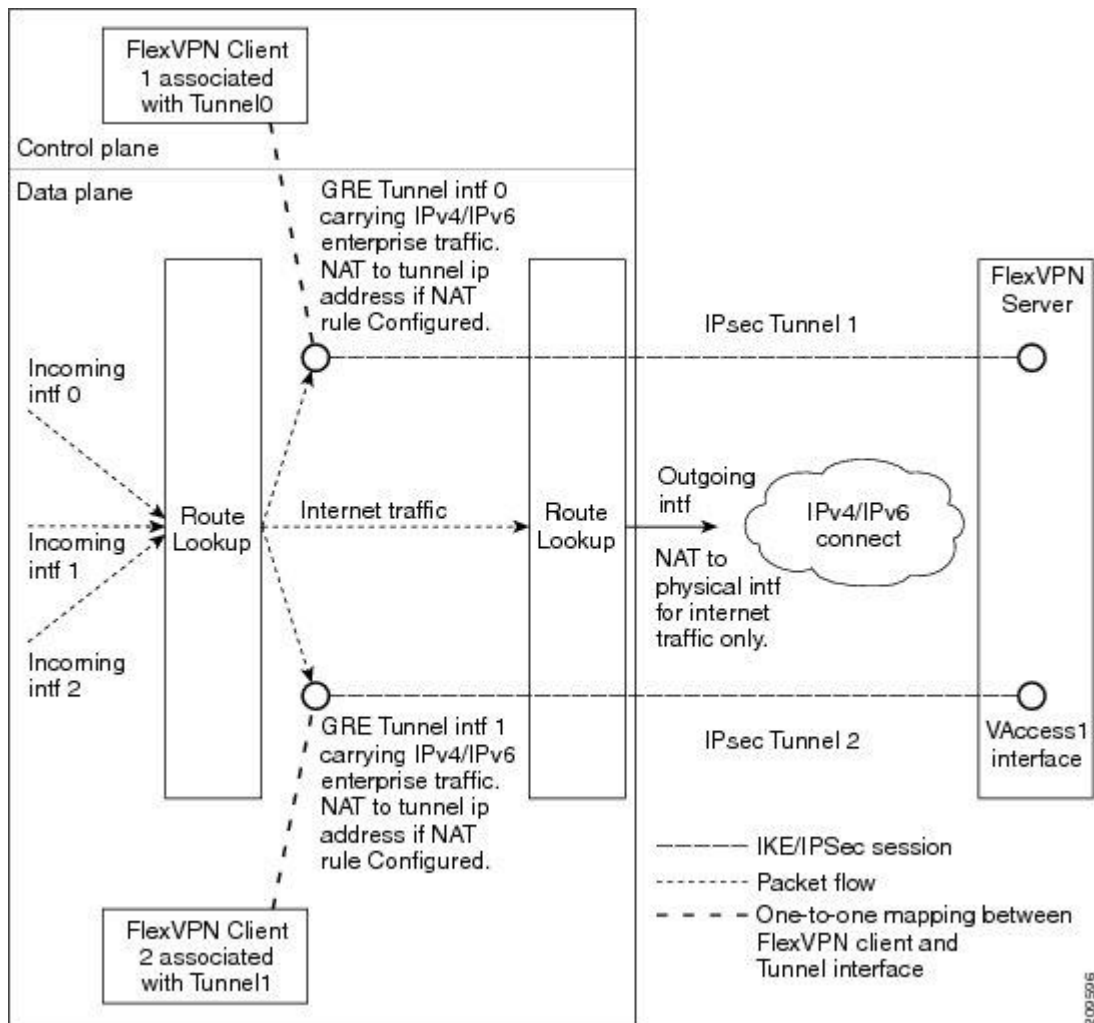


Figure 4. Association of the FlexVPN Client and the Tunnel Interface [4]

The sequence of operations is as follows:

- **Routing:** The FlexVPN server pushes the network list as part of the mode configuration response. The client adds routes on the tunnel interface to these networks. As part of the configuration mode set, the client sends the routes to its network. The IP address is configured on the tunnel interface so that the server can add routes to the client-side network.
- **Network Address Translation (NAT):** NAT rules must be configured explicitly using route maps. If the rules match, the hosts behind the FlexVPN client are translated to the tunnel IP address. This IP address can be obtained as one of the attributes pushed during mode configuration by the FlexVPN server.

- Encapsulation and encryption: Generic routing encapsulation (GRE) and IPsec encapsulation modes are supported. GRE supports both IPv4 and IPv6 traffic. The traffic that reaches the tunnel interface is encapsulated by the GRE header, followed by IPsec protection. The encrypted traffic is then routed to the outgoing interface.

The FlexVPN client can be connected automatically or manually through user intervention. The FlexVPN clients connects automatically to the tunnel when the FlexVPN configuration is complete. In a manual connection, the FlexVPN clients waits for user intervention to execute a command before establishing a connection.

- Configuring the Tunnel Interface

To configure the tunnel interface, create a tunnel interface and enter the interface configuration mode. Then assign an IPv4 address, enable GRE (Generic route encapsulation) and enable IPsec encapsulation. Specify the source, destination for the tunnel interface and associate a tunnel with an IPsec Profile.

- Configuring the FlexVPN Client

To configure the FlexVPN client, define an IKEv2 FlexVPN client profile and enter its configuration mode. Then define a static peer using an IP address or hostname and connect the FlexVPN tunnel, assign the tunnel interface created and add sequence number to the tunnel source address. After that, enable the reactivate primary peer feature and assign the client to a backup group.

- Configuring the EAP as the Local Authentication Method

To configure eap as the local authentication method enter IKEv2 profile configuration mode and specify eap as the local authentication method.

5 GET VPN

5.1 GET VPN Overview

In today's world there are various applications such as voice, video and some web-based applications in the network, and these applications increase the need for instantaneous, branch-interconnected, and QoS-enabled WANs. These applications are disseminated in the network, which results in increase demands for security and scale. Simultaneously, in order to trade-off between transport security and QoS-enabled branch interconnectivity, enterprise WAN technologies need their businesses. A next generation WAN encryption technology called GET VPN is introduced to reduce the network security risks, and it extinguish compromise that was made between network intellectualism and the privacy of the data.

After the introduction of this tunnel-less VPN called GET VPN the need for the tunnel was unnecessary. GET is a standards-based security model which is based on the idea of "trusted" group members. The security is very critical to the voice and video applications. Trusted members routers use this simple security method which does not depend on any point-to-point IPSec tunnel relationship. Furthermore, it uses trusted groups instead of point-to-point tunnels. Thus, the security level can scale higher in network-intelligence features while maintaining.

The network based on GET can be used in lots of WAN environments, including MPLS and IP. GET has a flexible nature. So, enterprise with security alert can conduct their own network or they can transfer encrypted services to their providers. Partial or full-mesh connectivity is required by large Layer 2 or MPLS networks. GET can simplify securing these large networks. [7]

GET VPN has various benefits and some of them are:

- It provides data security and transport authentication
- It helps to meet internal regulation and security compliance.
- Network intelligence, for example full-mesh connectivity, natural routing path and QoS for MPLS networks are maintained.
- Any to any instant enterprise connectivity that is optimal for voice over VPN deployments.

- It allows higher scalability and simplifies the troubleshooting
- It reduces traffic loads and by utilizing the network's core part for replication of multicast traffic, it provides edge encryption devices.

5.2 GET VPN Architecture

GET VPN encloses Multicast Rekeying, it is a way to enable encryption for native multicast packets, and unicast rekeying over a private WAN. As defined in Internet Engineering Task Force (IETF), Multicast Rekeying and GET VPN are based on Group Domain of Interpretation GDOI. It has got similarities with IPsec in the area of header preservation and SA lookup. The properties of IPsec, tunnel overlay has been removed and the dynamic distribution of IPsec SAs has been added. [7]

The figure below further illustrates the GET VPN concepts and relationships.

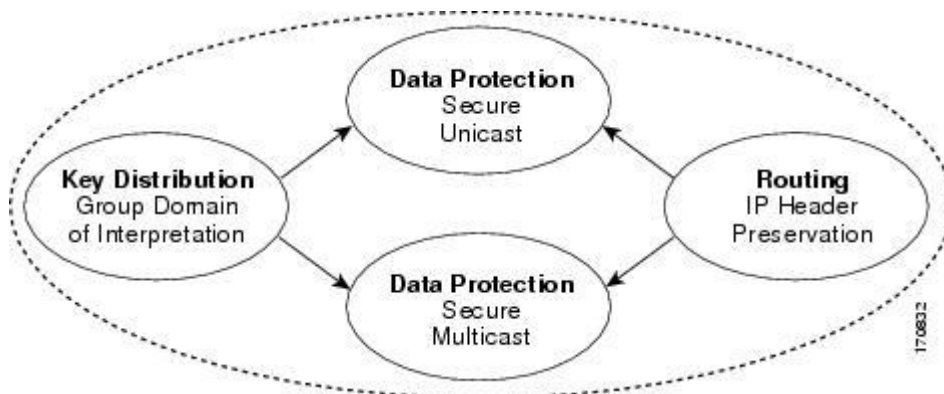


Figure 5. GET VPN Concepts and Relationships. [7]

5.2.1 Key Distribution Group Domain Interpretation

The ISAKMP Domain of Interpretation (DOI) for a group key management is GDOI (Group Domain Interpretation). This protocol set up security association between authorized group members while operating between group member and group controller.

IPSec security associations, it is very important for it to advertise with the group so the group member registers with the key server and arrange the group ID to the key server to get the IPSec security associations and to obtain the respective policy and the keys for this group respectively. Sustaining the policy, creating and sustaining the keys for the group are the responsibility of key server, it also rekeys the group before the existing key gets expired. After a group member has done its registration, the key server downloads the maintained policy and keys to the group.

After the group members register itself with the key server, the key server verifies and approves the group members and download the IPSec policy and keys that are essential for them to encode and decode IP multicast packets. A rekey message is pushed towards the group member by the key server. That rekey contains a new IPSec policy and key to utilize when old IPSec SAs expire. In order to guarantee that the valid group keys are always available, Rekeys message are sent ahead of time of the SA expiration time. The key server authenticates the group members and communicates with other authenticated group members that are in the same group and that are using the IPsec SAs that they received from the key server.

To distribute the policy and keys for the group, Multicast Rekeying utilizes the GDOI protocol which is situated between a key server and a group member. The key server transfers the policy and keys to the validated group members, it creates and maintains the policy and keys. An ISAKMP Phase 1 exchange protects the GDOI protocol. The GDOI key server and the GDOI group member should have the same ISAKMP policy. To secure the GDOI protocol that follows, the phase 1 ISAKMP policy must be very solid. The GDOI protocol is a four-message interchange that follows the Phase 1 ISAKMP policy. The exchange of Phase 1 ISAKMP can happen in aggressive mode or main mode. [7]

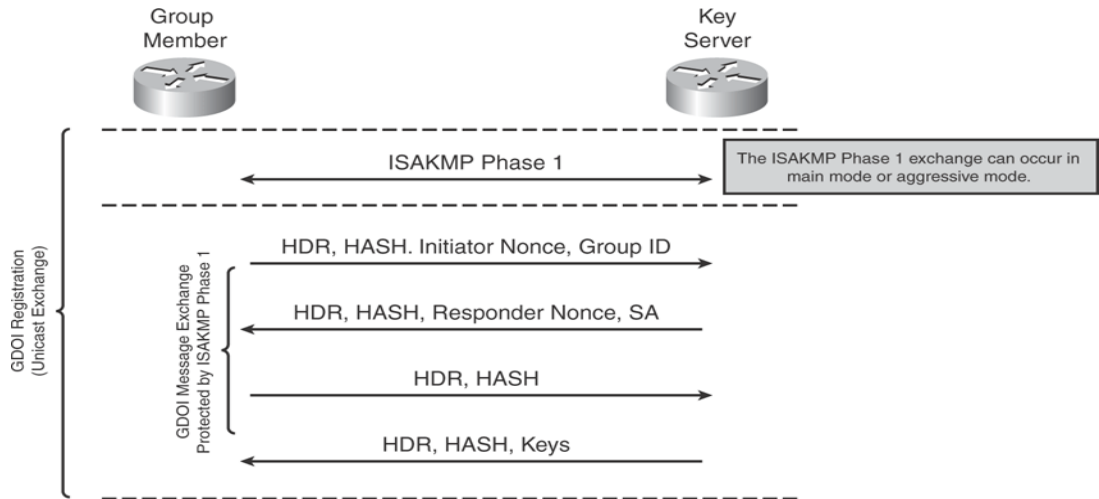


Figure 6: ISAKMP Phase 1 Exchange and GDOI Registration

As shown in the above Figure 6, the entire GODI registration procedure includes ISAKMP phase 1 message and the four GDOI protocol messages which is secured by ISAKMP phase 1.

5.2.2 Address Preservation

One of the main advantage of utilizing GET VPN is that it offers header or address observation. Address Preservation allows GET VPN to utilize its routing function and permits routing to deliver the packets to the end client device in the network that advertise a route to the destination address. End host address can be exposed in the WAN because the header preservation maintains the routing continuity in the WAN and enterprise address space. [7]

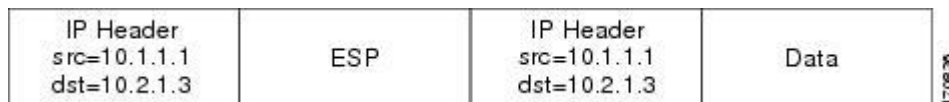


Figure 7. GET VPN Concepts and Relationships

As seen in the figure, the data packets carry the original source address and destination address in the IP header instead of using tunnel endpoint address. This method is the Address Preservation.

5.2.3 Secure Data Plane Multicast

The Traffic Encryption Key (TEK) is obtained from the key server. The multicast data packet that has the header preservation are encrypted by multicast using the TEK. After the encryption process the multicast changes the packet. Based on (S, G) state, the multicast packet is replicated in the core. [7]

5.2.4 Secure Data Plane Unicast

The Traffic Encryption Key (TEK) is obtained from the key server. The unicast data packet that has the header preservation are encrypted by unicast sender using the TEK. After the encryption process the unicast sender transfer the packet to the destination. [7]

6 FlexVPN Deployment

6.1 Requirements and Topology

The objective of this project is to establish a Site-to-Site FlexVPN between Branch 1 and Branch 2. Cisco 2911 router with Cisco IOS Release 15.7 is used during this project. For the implementation of FlexVPN, there must be a minimum hardware and software requirements to support FlexVPN. The prerequisites are as follows:

- Three Routers (Cisco 2911 with Cisco IOS Release 15.7)
- Two PCs
- Ethernet cables and Console cables to configure Cisco networking devices.

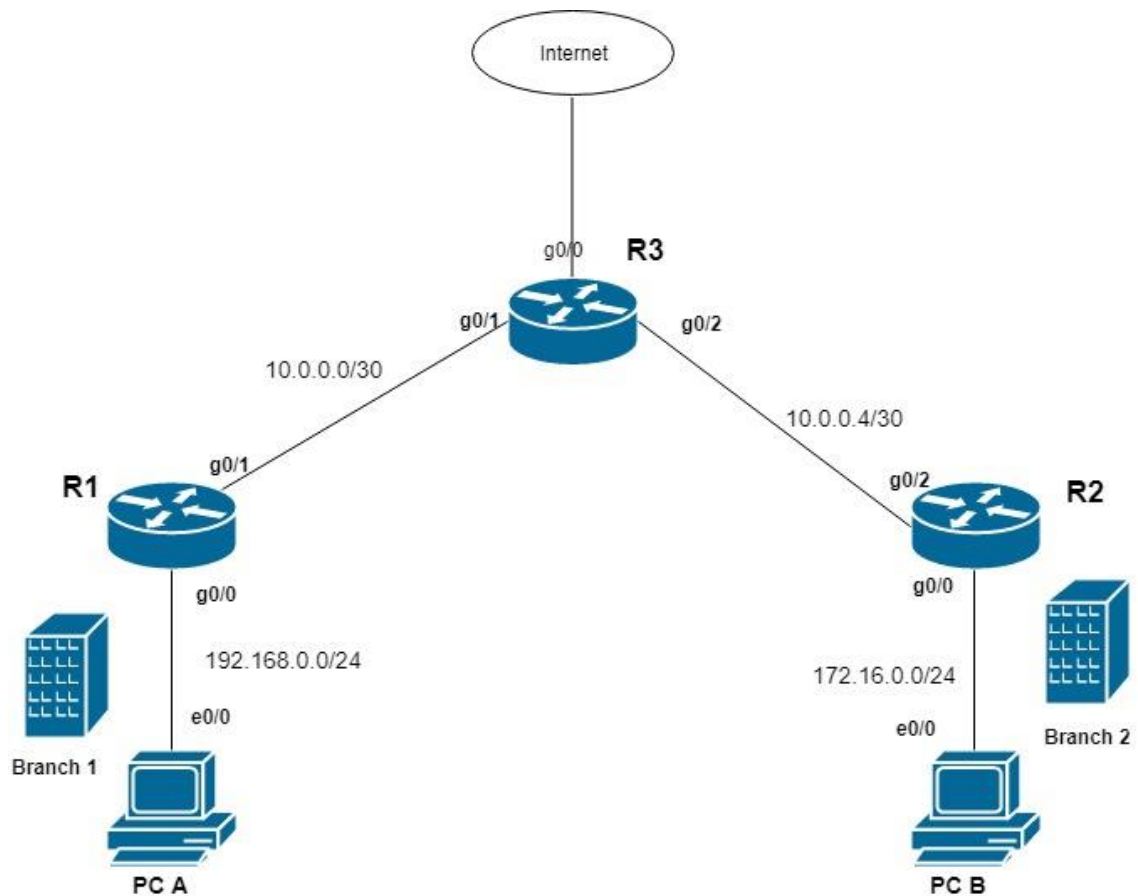


Figure 8. Configuration

The Project topology shown in the figure 8 is designed by considering the features of a very small company. The topology has two branch offices connected with each other from a Service point that connects them to the Main Building or the Internet.

Table 3: IP Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	g 0/0	192.168.0.254	255.255.255.0	N/A
	g 0/1	10.0.0.1	255.255.255.252	N/A
	Lo0	1.1.1.1	255.255.255.255	
R2	g 0/0	172.16.0.254	255.255.255.0	N/A
	g 0/2	10.0.0.5	255.255.255.252	N/A
	Lo0	2.2.2.2	255.255.255.255	
R3	g 0/0	N/A	N/A	N/A
	g 0/1	10.0.0.2	255.255.255.252	N/A
	g 0/2	10.0.0.6	255.255.255.252	N/A
PC A	NIC	192.168.0.1	255.255.255.0	192.168.0.254
PC B	NIC	172.16.0.1	255.255.255.0	172.16.0.254

6.2 Configuration

During the project, the network topology was set up and all the devices were attached with the cables as shown in the topology. The basic settings such as hostname configuration, IP address configuration were configured as shown in the IP Addressing Table. The further steps are mentioned below.

Step 1: Configure DHCP server and a domain name on both branches.

```
Branch1(config)#ip dhcp pool LAN
Branch1(dhcp-config)#network 192.168.0.0 255.255.255.0
Branch1(dhcp-config)# default-router 192.168.0.254
Branch1(dhcp-config)# dns-server 8.8.8.8
Branch1(dhcp-config)# domain-name thesisflex.net
Branch1(config)#ip domain name thesisflex.net
```

```
Branch2(config)#ip dhcp pool LAN
```

```
Branch2(dhcp-config)# network 172.16.0.0 255.255.255.0
Branch2(dhcp-config)# default-router 172.16.0.254
Branch2(dhcp-config)# dns-server 8.8.8.8
Branch2(dhcp-config)# domain-name thesisflex.net
Branch2(config)#ip domain name thesisflex.net
```

Step 2: Configure the IKEv2 Proposal on both branches.

```
Branch1(config)#crypto ikev2 proposal IKEPROPOSAL
Branch1(config-ikev2-proposal)# encryption aes-cbc-256
Branch1(config-ikev2-proposal)# integrity sha512
Branch1(config-ikev2-proposal)# group 14
```

```
Branch2(config)#crypto ikev2 proposal IKEPROPOSAL
Branch2(config-ikev2-proposal)# encryption aes-cbc-256
Branch2(config-ikev2-proposal)# integrity sha512
Branch2(config-ikev2-proposal)# group 14
```

Step 3: Configure the IKEv2 Policy with the same IKEv2 proposal name as above.

```
Branch1(config)#crypto ikev2 policy IKEPOLICY
Branch1(config-ikev2-policy)# match address local 10.0.0.1
Branch1(config-ikev2-policy)# proposal IKEPROPOSAL
```

```
Branch2(config)#crypto ikev2 policy IKEPOLICY
Branch2(config-ikev2-policy)# match address local 10.0.0.5
Branch2(config-ikev2-policy)# proposal IKEPROPOSAL
```

Step 4: Create a keyring, use fqdn identity for Branch 1 and email identity for Branch 2.

```
Branch1(config)#crypto ikev2 keyring KEYRING
Branch1(config-ikev2-keyring)# peer Branch2
Branch1(config-ikev2-keyring-peer)# description Branch2 PSK Authentication
Branch1(config-ikev2-keyring-peer)# address 10.0.0.5
Branch1(config-ikev2-keyring-peer)# identity fqdn
Branch2.thesisflex.net
Branch1(config-ikev2-keyring-peer)# pre-shared-key local Branch1key
Branch1(config-ikev2-keyring-peer)# pre-shared-key remote Branch2key
```

```

Branch2(config)#crypto ikev2 keyring KEYRING
Branch2(config-ikev2-keyring)# peer Branch1
Branch2(config-ikev2-keyring-peer)# description Branch1 PSK Authentication
Branch2(config-ikev2-keyring-peer)# address 10.0.0.1
Branch2(config-ikev2-keyring-peer)#          identity          email
branch1@thesisflex.net
Branch2(config-ikev2-keyring-peer)# pre-shared-key local Branch2key
Branch2(config-ikev2-keyring-peer)# pre-shared-key remote Branch1key

```

Step 5: Create a IKEv2 Profile matching the remote and local identity of both branches.

```

Branch1(config)#crypto ikev2 profile IKEPROFILE
Branch1(config-ikev2-profile)#$match          identity          remote          fqdn
Branch2.thesisflex.net
Branch1(config-ikev2-profile)#          identity          local          email
branch1@thesisflex.net
Branch1(config-ikev2-profile)# authentication remote pre-share
Branch1(config-ikev2-profile)# authentication local pre-share
Branch1(config-ikev2-profile)# keyring local KEYRING

Branch2(config)#crypto ikev2 profile IKEPROFILE
Branch2(config-ikev2-profile)#$match          identity          remote          email
branch1@thesisflex.net
Branch2(config-ikev2-profile)#          identity          local          fqdn
Branch2.thesisflex.net
Branch2(config-ikev2-profile)# authentication remote pre-share
Branch2(config-ikev2-profile)# authentication local pre-share
Branch2(config-ikev2-profile)# keyring local KEYRING

```

Step 6: Create a secure transform set and establish a tunnel between the branches.

```

Branch1(config)#crypto ipsec transform-set TSET esp-aes 256 esp-
sha512-hmac
Branch1(cfg-crypto-trans)# mode tunnel
Branch1(cfg-crypto-trans)#crypto ipsec profile IPSECPROFILE
Branch1(ipsec-profile)# set transform-set TSET

```

```
Branch1(ipsec-profile)# set ikev2-profile IKEPROFILE
Branch1(config)#interface Tunnell
Branch1(config-if)# ip unnumbered Loopback0
Branch1(config-if)# tunnel source GigabitEthernet0/1
Branch1(config-if)# tunnel mode ipsec ipv4
Branch1(config-if)# tunnel destination 10.0.0.5
Branch1(config-if)# tunnel path-mtu-discovery
Branch1(config-if)# tunnel protection ipsec profile IPSECPROFILE

Branch2(config)#crypto ipsec transform-set TSET esp-aes 256 esp-
sha512-hmac
Branch2(cfg-crypto-trans)# mode tunnel
Branch2(cfg-crypto-trans)#crypto ipsec profile IPSECPROFILE
Branch2(ipsec-profile)# set transform-set TSET
Branch2(ipsec-profile)# set ikev2-profile IKEPROFILE
Branch2(config)#interface Tunnell
Branch2(config-if)# ip unnumbered Loopback0
Branch2(config-if)# tunnel source GigabitEthernet0/2
Branch2(config-if)# tunnel mode ipsec ipv4
Branch2(config-if)# tunnel destination 10.0.0.1
Branch2(config-if)# tunnel path-mtu-discovery
Branch2(config-if)# tunnel protection ipsec profile IPSECPROFILE
```

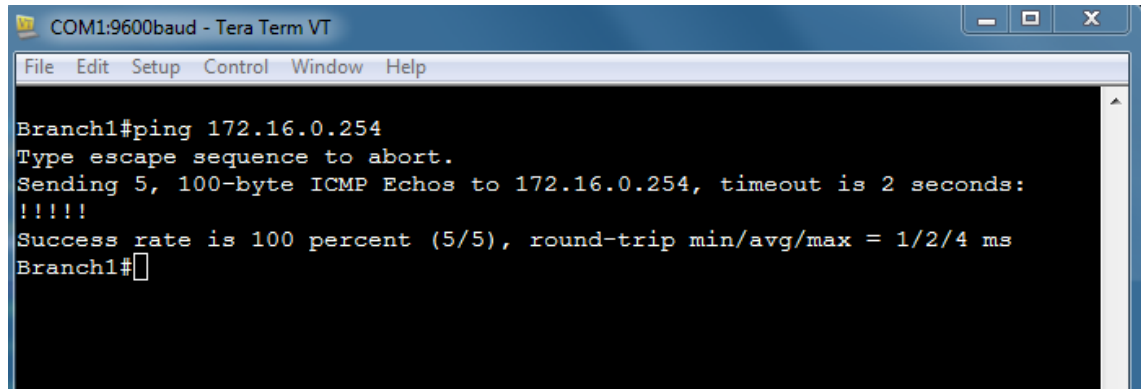
Step 7: Configure EIGRP routing protocol on both the branches.

```
Branch1(config)#router eigrp 10
Branch1(config-router)# network 1.1.1.1 0.0.0.0
Branch1(config-router)# network 192.168.0.254 0.0.0.0

Branch2(config)#router eigrp 10
Branch2(config-router)# network 2.2.2.2 0.0.0.0
Branch2(config-router)# network 172.16.0.254 0.0.0.0
```

6.3 Results

After the configuration on both branch and the Internet router, ping command was used to verify the end to end connection and show ip interfaces brief command was used to check the status of the interfaces and the tunnel.

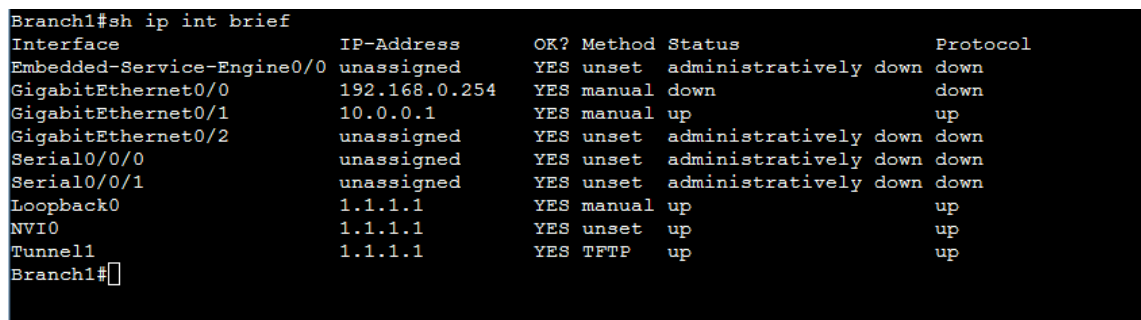


```

COM1:9600baud - Tera Term VT
File Edit Setup Control Window Help
Branch1#ping 172.16.0.254
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.0.254, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
Branch1#

```

Figure 9: Screenshot of Ping result

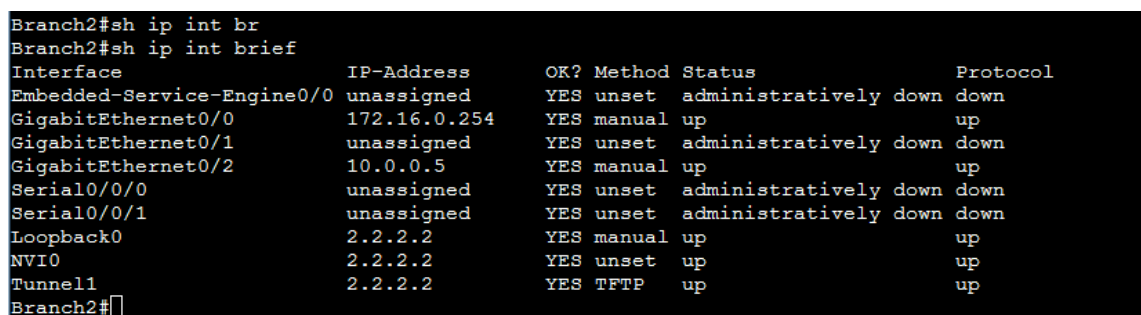


```

Branch1#sh ip int brief
Interface IP-Address OK? Method Status Protocol
Embedded-Service-Engine0/0 unassigned YES unset administratively down down
GigabitEthernet0/0 192.168.0.254 YES manual down down
GigabitEthernet0/1 10.0.0.1 YES manual up up
GigabitEthernet0/2 unassigned YES unset administratively down down
Serial0/0/0 unassigned YES unset administratively down down
Serial0/0/1 unassigned YES unset administratively down down
Loopback0 1.1.1.1 YES manual up up
NVI0 1.1.1.1 YES unset up up
Tunnel1 1.1.1.1 YES TFTP up up
Branch1#

```

Figure 10: Screenshot of Branch 1 showing interfaces status



```

Branch2#sh ip int br
Branch2#sh ip int brief
Interface IP-Address OK? Method Status Protocol
Embedded-Service-Engine0/0 unassigned YES unset administratively down down
GigabitEthernet0/0 172.16.0.254 YES manual up up
GigabitEthernet0/1 unassigned YES unset administratively down down
GigabitEthernet0/2 10.0.0.5 YES manual up up
Serial0/0/0 unassigned YES unset administratively down down
Serial0/0/1 unassigned YES unset administratively down down
Loopback0 2.2.2.2 YES manual up up
NVI0 2.2.2.2 YES unset up up
Tunnel1 2.2.2.2 YES TFTP up up
Branch2#

```

Figure 11: Screenshot of Branch 2 showing interfaces status

As can be seen in Figure 10 and Figure 11, tunnel 1 and all the configured interfaces has the status “up”.

After the configuration, show `crypto ikev2 sa` detailed command was used in both branch router to verify the connection and configuration.

```
Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.0.1/500 10.0.0.5/500 none/none READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:14, Auth sign: PSK, Auth verify: PSK
Life/Active Time: 86400/723 sec
CE id: 1002, Session-id: 1
Status Description: Negotiation done
Local spi: 98EF127F48AC6942 Remote spi: FEF7A03C01AA879E
Local id: branch1@thesisflex.net
Remote id: Branch2.thesisflex.net
Local req msg id: 2 Remote req msg id: 0
Local next msg id: 2 Remote next msg id: 0
Local req queued: 2 Remote req queued: 0
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
Fragmentation not configured.
Dynamic Route Update: disabled
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : Yes

IPv6 Crypto IKEv2 SA
```

Figure 12: Screenshot of Branch 1 showing detailed security association

```
Branch2#sh crypto ikev2 sa detailed
IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
2 10.0.0.5/500 10.0.0.1/500 none/none READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:14, Auth sign: PSK, Auth verify: PSK
Life/Active Time: 86400/558 sec
CE id: 1002, Session-id: 1
Status Description: Negotiation done
Local spi: FEF7A03C01AA879E Remote spi: 98EF127F48AC6942
Local id: Branch2.thesisflex.net
Remote id: branch1@thesisflex.net
Local req msg id: 0 Remote req msg id: 2
Local next msg id: 0 Remote next msg id: 2
Local req queued: 0 Remote req queued: 2
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
Fragmentation not configured.
Dynamic Route Update: disabled
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : No

IPv6 Crypto IKEv2 SA
```

Figure 13: Screenshot of Branch 2 showing detailed security association

As can be seen in Figure 12 and Figure 13, the tunnel between the two branch has the status “Ready” and has the remote id, local id, encryption, integrity, group and authentication as defined in the configuration section.

7 Conclusion and Discussion

The main goal of this project was to study the detailed configuration and deployment methods of Internet Key Exchange Version 2(IKEv2), FlexVPN and Group Encrypted Transport VPN (GET VPN) so that any reader can test these VPNs in real environment and evaluate and analyze the comparison between these technologies. The objective was also to deploy the FlexVPN in real environment in the laboratory for the testing purpose.

After the study of configuration and deployments methods, a network was built to accommodate the most possible scenario in reality at the laboratory. The IKEv2 Profile, policy, proposal and keyring were configured as per the requirements. After implementing everything, network that was established was tested, and it performed as per it was designed. The tunnel is implemented between the routers and the connection was successfully established, the tunnel and network traffic worked out as designed.

There were limitations in the project, it was not possible to implement a large network in the lab environment. So, the network that was designed during the project reflects a very small size organization network. But the methods of implementation that are used during this project are also useful to setup a network of any size. The main goal was achieved and in the process a VPN was created which is a useful roadmap for anyone interested in establishing a new VPN or testing these VPN networks.

References

- [1] Eric F Crist, Jan Just Keijser. Mastering OpenVPN [Online]. Available: <http://proquestcombo.safaribooksonline.com.ezproxy.metropolia.fi/book/networking/vpn/9781783553136> . Accessed 25th March 2018.
- [2] Thomas Berger. Analysis of current VPN Technologies. [Online]. Available: <http://ieeexplore.ieee.org.ezproxy.metropolia.fi/xpls/icp.jsp?arnumber=1625300>. Accessed 20th March 2018.
- [3] Graham Bartlett, Amjad Inamdar. IKEv2 IPsec VPNs, Understanding and Deploying IKEv2, IPsec VPNs and Flex VPN in Cisco IOS. Accessed 1st January 2018.
- [4] Configuring Internet Key Exchange Version 2(IKEv2) [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios/sec_secure_connectivity/configuration/guide/convert/sec_ike_for_ipsec_vpns_15_1_book/sec_cfg_ikev2.pdf. Accessed 14th January 2018
- [5] FlexVPN and Internet Key Exchange Version 2 Configuration Guide, Cisco IOS XE Release 3S [online]. Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_ike2vpn/configuration/15-mt/sec-flex-vpn-15-mt-book/sec-cfg-ikev2-flex.html. Accessed 24th December 2017.
- [6] Cisco IOS Flex VPN Datasheet. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/routers/asr-1000-series-aggregation-services-routers/data_sheet_c78-704277.html .Accessed 15th December 2017
- [7] Cisco Group Encrypted Transport VPN Configuration Guide, Cisco IOS XE Release 3S. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_conn_getvpn/configuration/xs-3s/sec-get-vpn-xe-3s-book/sec-get-vpn.html#GUID-6879A26D-A555-4DE8-97C6-91564ACC8ACA . Accessed 24th January 2018
- [8] Top9 Tips, VPN providers review. [Online]. Available: <http://top9tips.com/top-3-vpn-providers-review/> . Accessed 20th February 2018

Appendix 1. Branch 1 Running Configuration

```
Branch1#sh run
Building configuration...

Current configuration : 3176 bytes
!
! Last configuration change at 15:22:02 UTC Tue Mar 27 2018
!
version 15.7
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Branch1
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
memory-size iomem 5
!
!
!
!
!
!
!
!
!
!
ip dhcp pool LAN
 network 192.168.0.0 255.255.255.0
 default-router 192.168.0.254
 dns-server 8.8.8.8
 domain-name thesisflex.net
!
!
!
ip domain name thesisflex.net
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
license udi pid CISCO2911/K9 sn FCZ1545206V
license accept end user agreement
license boot module c2900 technology-package securityk
```

```
license boot module c2900 technology-package uck9 disable
!
!
!
redundancy
!
crypto ikev2 proposal IKEPROPOSAL
  encryption aes-cbc-256
  integrity sha512
  group 14
!
crypto ikev2 policy IKEPOLICY
  match address local 10.0.0.1
  proposal IKEPROPOSAL
!
crypto ikev2 keyring KEYRING
  peer Branch2
    description Branch2 PSK Authentication
    address 10.0.0.5
    identity fqdn Branch2.thesisflex.net
    pre-shared-key local Branch1key
    pre-shared-key remote Branch2key
!
!
!
crypto ikev2 profile IKEPROFILE
  match identity remote fqdn Branch2.thesisflex.net
  identity local email branch1@thesisflex.net
  authentication remote pre-share
  authentication local pre-share
  keyring local KEYRING
!
!
!
!
!
crypto ipsec transform-set TSET esp-aes 256 esp-sha512-hmac
  mode tunnel
!
crypto ipsec profile IPSECPROFILE
  set transform-set TSET
  set ikev2-profile IKEPROFILE
!
!
!
!
!
!
interface Loopback0
  ip address 1.1.1.1 255.255.255.255
!
interface Tunnell
```

```
ip unnumbered Loopback0
tunnel source GigabitEthernet0/1
tunnel mode ipsec ipv4
tunnel destination 10.0.0.5
tunnel path-mtu-discovery
tunnel protection ipsec profile IPSECPROFILE
!
interface Embedded-Service-Engine0/0
no ip address
shutdown
!
interface GigabitEthernet0/0
description LAN CONNECTION
ip address 192.168.0.254 255.255.255.0
ip nat inside
ip virtual-reassembly in
duplex auto
speed auto
!
interface GigabitEthernet0/1
description INTERNET CONNETION
ip address 10.0.0.1 255.255.255.252
ip nat outside
ip virtual-reassembly in
duplex auto
speed auto
!
interface GigabitEthernet0/2
no ip address
shutdown
duplex auto
speed auto
!
interface Serial0/0/0
no ip address
shutdown
clock rate 2000000
!
interface Serial0/0/1
no ip address
shutdown
clock rate 2000000
!
!
router eigrp 10
network 1.1.1.1 0.0.0.0
network 192.168.0.254 0.0.0.0
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip nat inside source list NAT interface GigabitEthernet0/1 overload
ip route 0.0.0.0 0.0.0.0 10.0.0.2
```

```
ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr
ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr
!
ip access-list extended NAT
  permit ip 192.168.0.0 0.0.0.255 any
  deny ip any any
!
!
!
!
control-plane
!
!
  vstack
!
line con 0
line aux 0
line 2
  no activation-character
  no exec
  transport preferred none
  transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
  stopbits 1
line vty 0 4
  login
  transport input none
!
scheduler allocate 20000 1000
!
End
```

Appendix 2. Branch 2 Running Configuration

```
Branch2#sh run
Building configuration...

Current configuration : 3172 bytes
!
! Last configuration change at 15:42:43 UTC Tue Mar 27 2018
!
version 15.7
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Branch2
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
memory-size iomem 5
!
!
!
!
!
!
!
!
!
!
!
ip dhcp pool LAN
 network 172.16.0.0 255.255.255.0
 default-router 172.16.0.254
 dns-server 8.8.8.8
 domain-name thesisflex.net
!
!
!
ip domain name thesisflex.net
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
license udi pid CISCO2911/K9 sn FCZ15457077
```

```
license accept end user agreement
license boot module c2900 technology-package securityk9
license boot module c2900 technology-package uck9 disable
!
!
!
redundancy
!
crypto ikev2 proposal IKEPROPOSAL
  encryption aes-cbc-256
  integrity sha512
  group 14
!
crypto ikev2 policy IKEPOLICY
  match address local 10.0.0.5
  proposal IKEPROPOSAL
!
crypto ikev2 keyring KEYRING
  peer Branch1
    description Branch1 PSK Authentication
    address 10.0.0.1
    identity email branch1@thesisflex.net
    pre-shared-key local Branch2key
    pre-shared-key remote Branch1key
!
!
!
crypto ikev2 profile IKEPROFILE
  match identity remote email branch1@thesisflex.net
  identity local fqdn Branch2.thesisflex.net
  authentication remote pre-share
  authentication local pre-share
  keyring local KEYRING
!
!
!
!
!
crypto ipsec transform-set TSET esp-aes 256 esp-sha512-hmac
  mode tunnel
!
crypto ipsec profile IPSECPROFILE
  set transform-set TSET
  set ikev2-profile IKEPROFILE
!
!
!
!
!
!
interface Loopback0
  ip address 2.2.2.2 255.255.255.255
```



```
!  
interface Tunnell  
  ip unnumbered Loopback0  
  tunnel source GigabitEthernet0/2  
  tunnel mode ipsec ipv4  
  tunnel destination 10.0.0.1  
  tunnel path-mtu-discovery  
  tunnel protection ipsec profile IPSECPROFILE  
!  
interface Embedded-Service-Engine0/0  
  no ip address  
  shutdown  
!  
interface GigabitEthernet0/0  
  description LAN CONNECTION  
  ip address 172.16.0.254 255.255.255.0  
  ip nat inside  
  ip virtual-reassembly in  
  duplex auto  
  speed auto  
!  
interface GigabitEthernet0/1  
  no ip address  
  shutdown  
  duplex auto  
  speed auto  
!  
interface GigabitEthernet0/2  
  description INTERNET CONNETION  
  ip address 10.0.0.5 255.255.255.252  
  ip nat outside  
  ip virtual-reassembly in  
  duplex auto  
  speed auto  
!  
interface Serial0/0/0  
  no ip address  
  shutdown  
  clock rate 2000000  
!  
interface Serial0/0/1  
  no ip address  
  shutdown  
  clock rate 2000000  
!  
!  
router eigrp 10  
  network 2.2.2.2 0.0.0.0  
  network 172.16.0.254 0.0.0.0  
!  
ip forward-protocol nd  
!  
no ip http server  
no ip http secure-server  
!
```

```
ip nat inside source list NAT interface GigabitEthernet0/2 overload
ip route 0.0.0.0 0.0.0.0 10.0.0.6
ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr
ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr
!
ip access-list extended NAT
  permit ip 172.16.0.0 0.0.0.255 any
  deny ip any any
!
!
!
!
control-plane
!
!
  vstack
!
line con 0
line aux 0
line 2
  no activation-character
  no exec
  transport preferred none
  transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
  stopbits 1
line vty 0 4
  login
  transport input none
!
scheduler allocate 20000 1000
!
end
```

Appendix 3. Internet Running Configuration

```
INTERNET#sh run
Building configuration...

Current configuration : 2128 bytes
!
! Last configuration change at 15:49:50 UTC Tue Mar 27 2018
!
version 15.7
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname INTERNET
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
memory-size iomem 5
!
!
!
!
!
!
!
!
!
!
!
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
!
!
!
license udi pid CISCO2911/K9 sn FCZ15457078
license accept end user agreement
license boot module c2900 technology-package securityk9
license boot module c2900 technology-package uck9 disable
license boot module c2900 technology-package datak9 disable
!
!
```

```
redundancy
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
interface Embedded-Service-Engine0/0
  no ip address
  shutdown
!
interface GigabitEthernet0/0
  description INTERNET CONNECTION
  ip address dhcp
  ip nat outside
  ip virtual-reassembly in
  duplex auto
  speed auto
!
interface GigabitEthernet0/1
  description Branch1 CONNECTION
  ip address 10.0.0.2 255.255.255.252
  ip nat inside
  ip virtual-reassembly in
  duplex auto
  speed auto
!
interface GigabitEthernet0/2
  description Branch2 CONNECTION
  ip address 10.0.0.6 255.255.255.252
  ip nat inside
  ip virtual-reassembly in
  duplex auto
  speed auto
!
interface Serial0/0/0
  no ip address
  shutdown
  clock rate 2000000
!
interface Serial0/0/1
  no ip address
  shutdown
  clock rate 2000000
!
interface Serial0/1/0
```

```
no ip address
shutdown
clock rate 2000000
!
interface Serial0/1/1
no ip address
shutdown
clock rate 2000000
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip nat inside source list NAT interface GigabitEthernet0/0 overload
ip ssh server algorithm encryption aes128-ctr aes192-ctr aes256-ctr
ip ssh client algorithm encryption aes128-ctr aes192-ctr aes256-ctr
!
ip access-list extended NAT
permit ip 10.0.0.0 0.0.0.7 any
deny ip any any
!
!
!
!
control-plane
!
!
vstack
!
line con 0
line aux 0
line 2
no activation-character
no exec
transport preferred none
transport output pad telnet rlogin lapb-ta mop udptn v120 ssh
stopbits 1
line vty 0 4
login
transport input none
!
scheduler allocate 20000 1000
!
End
```

Appendix 4. IKEv2 Configurations

Configuring Global IKEv2 Options

```
enable
configure terminal
crypto ikev2 certificate-cache number-of-certificates
crypto ikev2 cookie-challenge number
crypto ikev2 diagnose error number
crypto ikev2 dpd interval retry-interval {on-demand | periodic}
crypto ikev2 http-url cert
crypto ikev2 limit {max-in-negotiation-sa limit | max-sa limit}
crypto ikev2 nat keepalive interval
crypto ikev2 window size
crypto logging ikev2
end
```

Configuring the IKEv2 Proposal

```
enable
configure terminal
crypto ikev2 proposal name
encryption {3des} {aes-cbc-128} {aes-cbc-192} {aes-cbc-256}
integrity {sha1} {sha256} {sha384} {sha512} {md5}
group {1} {2} {5} {14} {15} {16} {19} {20} {24}
end
show crypto ikev2 proposal [name| default]
```

Configuring the IKEv2 Policy

```
enable
configure terminal
crypto ikev2 policy name
proposal name
match fvrf {fvrf-name | any}
match address local { ipv4-address | ipv6-address}
end
show crypto ikev2 policy [policy-name]
```

- ### Configuring the IKEv2 Keyring

```
enable
configure terminal
crypto ikev2 keyring keyring-name
peer name
description line-of-description
hostname name
address {ipv4-address [mask] | ipv6-address prefix}
identity {address { ipv4-address | ipv6-address} | fqdn name | email
email-id | key-id key-id}
pre-shared-key {local| remote} {0| 6| line}
end
```

- Configuring the IKEv2 Profile

```
enable
configure terminal
crypto ikev2 profile profile-name
description line-of-description
aaa accounting [psk | cert | eap] list-name
aaa authentication eap list-name
authentication {local{rsa-sig| pre-share | ecdsa-sig} | remote{eap
[query-identity]| rsa-sig| pre-share | ecdsa-sig}
aaa authorization {group | user} [cert | eap | psk] aaa-listname {aaa-
username | name-mangler mangler-name}
dpd interval retry-interval {on-demand| periodic}
identity local {address { ipv4-address | ipv6-address}| dn| email
email-string| fqdn fqdn-string | key-id opaque-string}
ivrf name
keyring [aaa] name
lifetime seconds
match {address local{ipv4-address | ipv6-address}| interface name}|
certificate certificate-map | fvrf{fvrf-name | any}| identity re-
mote{address { ipv4-address [mask] | ipv6-address prefix} |
email[domain] string | fqdn[domain] string | key-id opaque-string}
nat keepalive seconds
pki trustpoint trustpoint-label [sign| verify]
virtual-template number
end
```

- Configuring the IKEv2 Name Mangler

```
enable
configure terminal
crypto ikev2 name-mangler mangler-name
dn {common-name | country | domain | locality | organization | organi-
zation-unit | state}
eap {all| dn {common-name | country | domain | locality | organization
| organization-unit | state} | {pre-fix | suffix {delimiter {. | @ |
\}}}}
email {all| domain | username}
fqdn {all| domain | hostname}
end
```

- Configuring the IKEv2 Authorization Policy

```
enable
configure terminal
crypto ikev2 authorization policy policy-name
dhcp {giaddr ip-address | server {ip-address | hostname} | timeout
seconds}
dns primary-server [secondary-server]
netmask mask
pool name
subnet-acl {acl-number | acl-name}
wins primary-server [secondary-server]
end
```

- Configuring the IKEv2 Fragmentation

```
enable
configure terminal
crypto ikev2 fragmentation [mtu mtu-size]
end
```


Appendix 5. FlexVPN Configurations

```
enable
configure terminal
crypto ikev2 profile profile-name
aaa authentication eap list-name
authentication {local {rsa-sig | pre-share [key {0 | 6} password]} |
ecdsa-sig | eap [gtc | md5 | ms-chapv2] [usernameusername] [pass-word
{0 | 6} password]}} | remote {eap [query-identity | timeout seconds] |
rsa-sig | pre-share [key {0 | 6}password]} | ecdsa-sig}}
Execute both or one of the following:
•      aaa authorization user {eap | psk} {cached | list aaa-
listname [aaa-username | name-mangler mangler-name]}
•      aaa authorization user cert list aaa-listname {aaa-username
| name-mangler mangler-name}
Execute both or one of the following:
•      aaa authorization group [override] {eap | psk} list aaa-
listname [aaa-username | name-mangler mangler-name]
•      aaa authorization group [override] cert list aaa-listname
{aaa-username | name-mangler mangler-name}
config-exchange {request | set {accept | send}}
end
```

- Configuring the IKEv2 Name Mangler

```
enable
configure terminal
crypto ikev2 name-mangler mangler-name
dn {common-name | country | domain | locality | organization
|organization-unit | state}
eap {all | dn {common-name | country | domain | locality |organization
| organization-unit | state} | prefix | suffix {delimiter {. | @| \}}}
email {all | domain | username}
fqdn {all | domain | hostname}
end
```

- Configuring the IKEv2 Authorization Policy

```
enable
configure terminal
crypto ikev2 authorization policy policy-name
aaa attribute list list-name
backup-gateway string
banner banner-text
configuration url url
configuration version version
def-domain domain-name
dhcp {giaddr ip-address | server {ip-address | hostname} | timeoutseconds}
[ipv6] dns primary-server [secondary-server]
include-local-lan
ipsec flow-limit number
netmask mask
pfs
[ipv6] pool name
route set {interface interface | access-list {access-list-name
|access-list-number | expanded-access-list-number | ipv6 access-list-
name}}
route accept any [tag value] [distance value]
route set remote {ipv4 ip-address mask | ipv6 ip-address/mask}
smartcard-removal-disconnect
split-dns string
session-lifetime seconds
route set access-list {acl-number | [ipv6] acl-name}
wins primary-server [secondary-server]
end
```

- Configuring the Tunnel Interface

```
enable
```

```
configure terminal
interface tunnel number
ip address {ipv4-address | negotiated}
tunnel mode gre ip
tunnel mode ipsec ipv4
tunnel source {ip-address | interface | dynamic}
tunnel destination dynamic
tunnel protection ipsec-profile profile-name
end
```

- Configuring the FlexVPN Client

```
enable
configure terminal
crypto ikev2 client flexvpn client-name
peer sequence {ipv4-address | ipv6-address | fqdn fqdn-name[dynamic |
ipv6]} [track track-number [up | down]]
connect {manual | auto | track track-number [up | down]}
client inside interface-type interface-number
client connect tunnel interface-number
source sequence-number interface-type interface-number tracktrack-
number
peer reactivate
backup group {group-number | default}
end
```

- Configuring the EAP as the Local Authentication Method

```
enable
configure terminal
crypto ikev2 profile profile-name
authentication local eap
end
```

Appendix 6. GET VPN Configurations

Configuring a Key Server

- Configuring RSA Keys to Sign Rekey Messages

```
enable
configure terminal
crypto key generate rsa general-keys label name-of-key
```

- Configuring the Group ID Server Type and SA Type

```
enable
configure terminal
crypto gdoi group group-name
Enter one of the following commands:
• identity number number
• identity address ipv4 address
server local
sa receive-only
```

- Configuring the Rekey

Configuring a Unicast Rekey

```
enable
configure terminal
crypto gdoi group group-name
Enter one of the following commands:
identity number number
identity address ipv4 address
server local
rekey transport unicast
rekey lifetime seconds number-of-seconds
rekey retransmit number-of-seconds number number-of-retransmissions
rekey authentication mypubkey rsa key-name
address                ipv4                ipv4-address
```

Configuring a Multicast Rekey

```

enable
configure terminal
crypto gdoi group group-name
Enter one of the following commands:
identity number number
identity address ipv4 address
server local
rekey address ipv4 {access-list-name | access-list-number}
rekey lifetime seconds number-of-seconds
rekey retransmit number-of-seconds number number-of-retransmissions
rekey authentication {mypubkey | pubkey} rsa key-name
exit
exit
access-list access-list-number {deny | permit} udp host
source[operator[port]] host source [operator[port]]
interface type slot/ port
ip igmp join-group group-address [source source-address]

```

- Configuring Group Members ACLs

```

enable
configure terminal
access-list access-list-number deny ip host source host source
access-list access-list-number permit ip source

```

- Configuring an IPsec Lifetime timer

```

enable
configure terminal
crypto ipsec profile name
set security-association lifetime seconds seconds

```

- Configuring an ISAKMP Lifetime Timer

```
enable
configure terminal
crypto isakmp policy priority
lifetime seconds
```

- Configuring the IPsec SA

```
enable
configure terminal
crypto ipsec transform-set transform-set-name transform [trans-
form2...transform4]
crypto ipsec profile ipsec-profile-name
set transform-set transform-set-name
exit
crypto gdoi group group-name
Enter one of the following commands:
identity number number
identity address ipv4 address
server local
sa ipsec sequence-number
profile ipsec-profile-name
match address ipv4 {access-list-number | access-list-name}
end
```

- Configuring Time-Based Antireplay for a GDOI Group

```
enable
configure terminal
crypto gdoi group group-name
identity number policy-name
server local
address ip-address
sa ipsec sequence-number
profile ipsec-profile-name
match address {ipv4 access-list-number | access-list-name}
replay counter window-size seconds
replay                time                window-size                seconds
```

- `Configuring Passive SA`
`enable`
`configure terminal`
`crypto gdoi group group-name`
`identity name`
`passive`
`server address ipv4 {address | hostname}`

5.3.2 `Configuring a Group Member`

We need to follow the following sub tasks to configure a group member.

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

```
enable
configure terminal
crypto gdoi group group-name
Do one of the following:
identity number number
identity address ipv4 address
server address ipv4 address
```

- `Configure a Crypto Map Entry`

```
enable
configure terminal
crypto map map-name seq-num gdoi
set group group-name
```

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

```
enable
configure terminal
interface type slot / port
crypto map map-name redundancy standby-group-name stateful
```

- `Activating Fail-Close Mode`

```
enable
```

```
configure terminal
crypto map map-name gdoi fail-close
match address {access-list-number | access-list-name}
activate
```

- Configuring Acceptable Ciphers or Hash Algorithms for KEK

```
enable
configure terminal
crypto gdoi group group-name
Enter one of the following commands:
identity number number
identity address ipv4 address
server address ipv4 address
client rekey encryption cipher [... [cipher]]
client rekey hash hash
end
```

- Configuring Acceptable Transform Sets for TEK

```
enable
configure terminal
crypto ipsec transform-set transform-set-name transform [trans-
form2...transform4]
exit
crypto gdoi group group-name
client transform-sets transform-set-name1 [... [transform-set-name6]]
end
```

- Tracking the Group Member Crypto State

```
enable
configure terminal
crypto gdoi group group-name
client status active-sa track tracking-number
exit
```


5.3.3 Configuring GET VPN GM Authorization

- Configuring GM Authorization Using the Preshared Keys

```
enable
configure terminal
crypto gdoi group group-name
server local
authorization address ipv4 {access-list-name | access-list-number}
exit
exit
access-list access-list-number [dynamic dynamic-name [timeoutminutes]]
{deny | permit} protocol source source-wildcard destinationdestina-
tion-wildcard [precedence precedence] [tos tos] [time-rangetime-range-
name] [frag-ments] [log [word] | log-input [word]]
exit
```

- Configuring GM Authorization Using PKI

```
enable
configure terminal
crypto isakmp identity {address | dn | hostname}
crypto pki trustpoint name
subject-name [x.500-name]
exit
crypto gdoi group group-name
server local
authorization identity name
exit
exit
crypto identity name
dn name=string [, name=string]
exit
crypto isakmp identity {address | dn | hostname}
crypto pki trustpoint name
subject-name [x.500-name]
end
```