BSc in Computer Engineering
CMP4103

Computer Systems and Network Security

Lecture 3a
Secure Virtual Private Networks
(Supplementary notes)

Eng Diarmuid O'Briain, CEng, CISSP

Department of Electrical and Computer Engineering,
College of Engineering, Design, Art and Technology,
Makerere University

Table of Contents

1 Virtual Private Networks

1.1 What is a Virtual Private Network

A virtual private network (VPN) is a computer network in which some of the links between nodes are carried by open connections or virtual circuits in some larger network (e.g., the Internet) instead of by physical wires. The link-layer protocols of the virtual network are said to be tunnelled through the larger network when this is the case. One common application is secure communications through the public Internet, but a VPN need not have explicit security features, such as authentication or content encryption. VPNs, for example, can be used to separate the traffic of different user communities over an underlying network with strong security features.

We will look at linking private LANs across a public network like the Internet. We will look at Generic Routing Encapsulation (GRE) and Internet Protocol Security (IPsec).

2 Generic Routing Encapsulation (GRE)

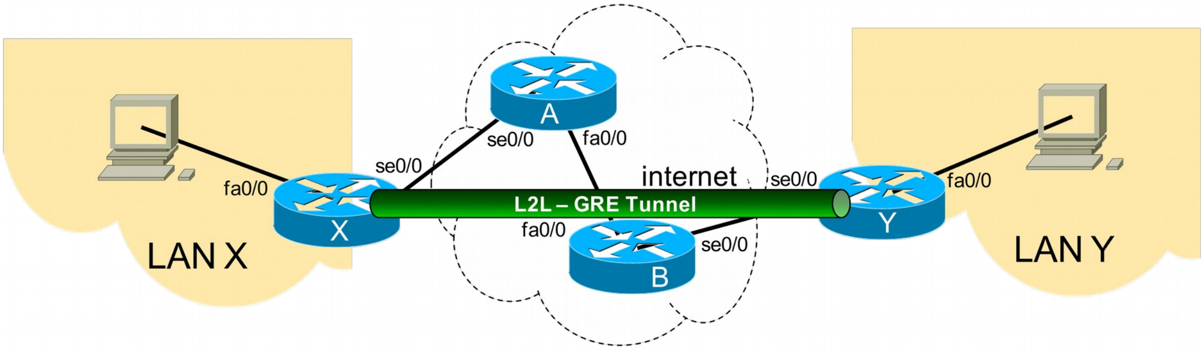
GRE is a tunnelling protocol developed by Cisco that can encapsulate a wide variety of network layer protocol packet types inside IP tunnels, creating a virtual point-to-point link to Cisco routers at remote points over an IP internetwork.

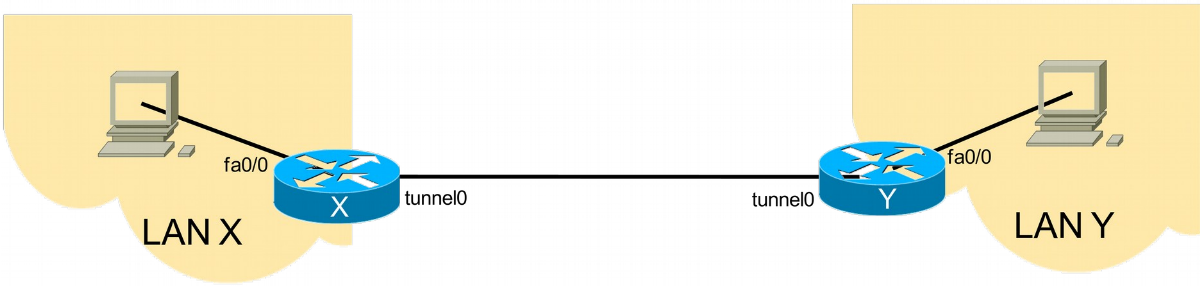
GRE is now a standard, defined in RFC 1701, RFC 1702, and RFC 2784.

GRE is a tunnelling protocol used to transport packets from one network through another network. It is a transport layer protocol with a protocol number of 47.

GRE tunnel is a type of a VPN but it is not a secure tunnelling method. However, you can encrypt GRE with an encryption protocol such as IPsec to form a secure VPN.

In the diagram below the routers A and B make up the public network with routers X and Y having interfaces on this public network. X and Y also have networks that are private and the objective is to allow these to connect in a GRE tunnel. From a routing perspective LAN Y appears to be only 1 hop from LAN X. The second picture is showing what in effect the private network sees

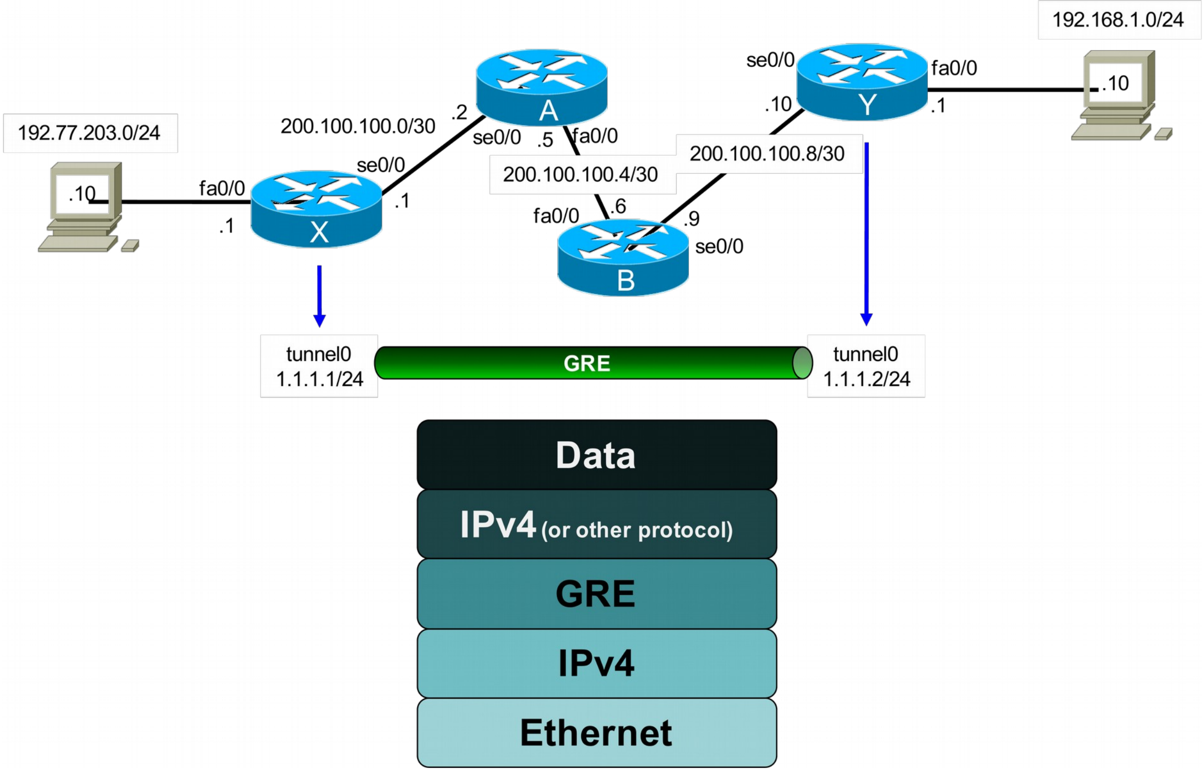


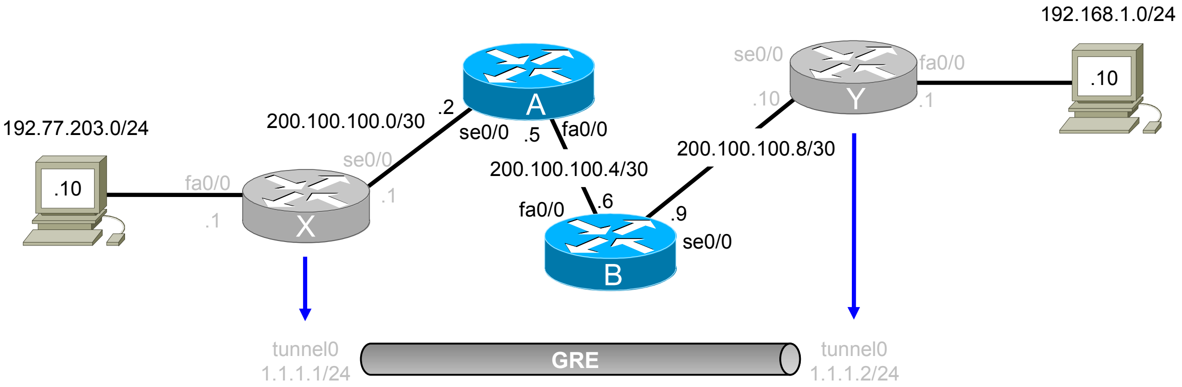


2.1 Configuring a GRE Tunnel

Here is a network to help demonstrate a GRE configuration. Routers A & B are public core routers and routers X & Y are access routers that connect remote sites to the Internet.

We can see from the model that GRE is carried by IPv4, within the GRE another IPv4 packet is carried.





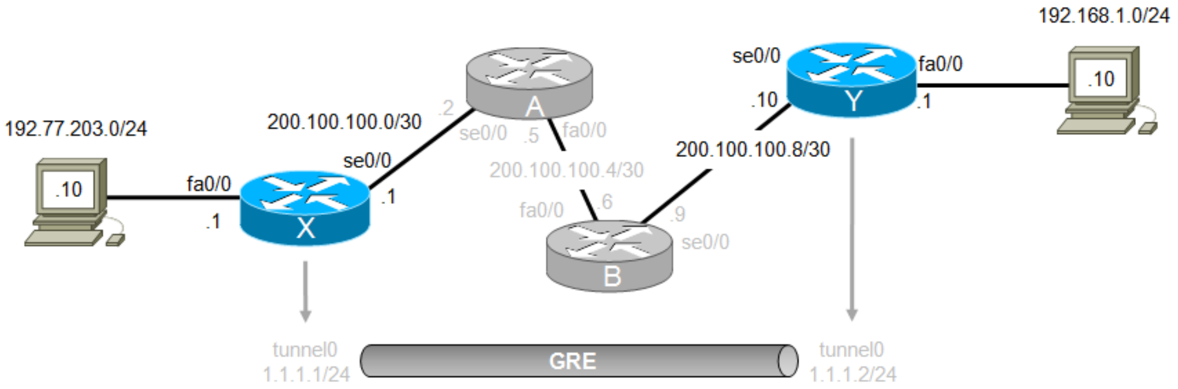
2.1.1 Initial configuration of the public core routers

Cisco A

```
Router(config)# hostname Cisco_A
Cisco_A(config)# ip routing
Cisco_A(config)# int fa 0/0
Cisco_A(config-if)# ip address 200.100.100.5 255.255.255.252
Cisco_A(config-if)# no shutdown
Cisco_A(config-if)# exit
Cisco_A(config)# int se 0/0
Cisco_A(config-if)# clock rate 72000
Cisco_A(config-if)# bandwidth 72
Cisco_A(config-if)# ip address 200.100.100.2 255.255.255.252
Cisco_A(config-if)# no shutdown
Cisco_A(config-if)# exit
Cisco_A(config)# router ospf 100
Cisco_A(config-router)# network 200.100.100.0 0.0.0.255 area 0.0.0.0
```

Cisco B

```
Router(config)# hostname Cisco_B
Cisco_B(config)# ip routing
Cisco_B(config)# int fa 0/0
Cisco_B(config-if)# ip address 200.100.100.6 255.255.255.252
Cisco_B(config-if)# no shutdown
Cisco_B(config-if)# exit
Cisco_B(config)# int se 0/0
Cisco_B(config-if)# clock rate 72000
Cisco_B(config-if)# bandwidth 72
Cisco_B(config-if)# ip address 200.100.100.9 255.255.255.252
Cisco_B(config-if)# no shutdown
Cisco_B(config-if)# exit
Cisco_B(config)# router ospf 100
Cisco_B(config-router)# network 200.100.100.0 0.0.0.255 area 0.0.0.0
```



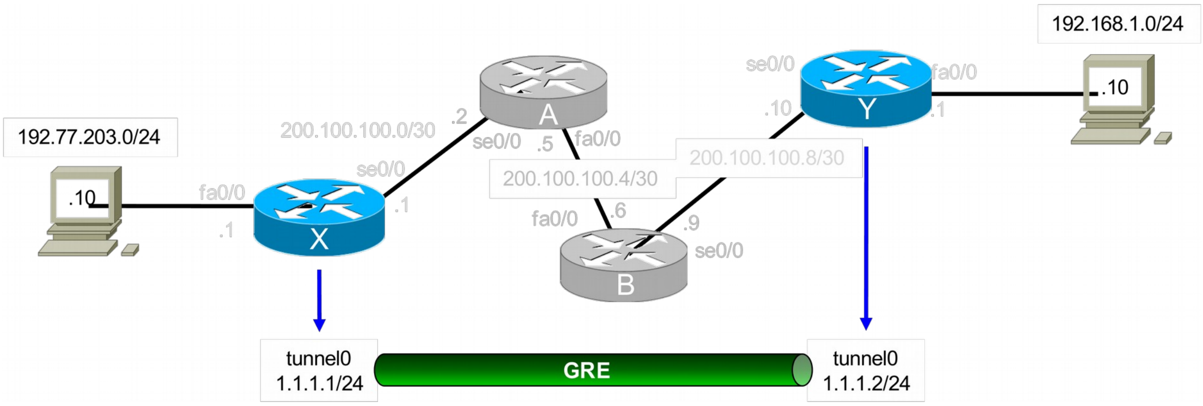
2.1.2 Initial configuration of the edge routers

Cisco X

```
Router(config)# hostname Cisco_X
(config)# ip routing
Cisco_X(config)# int fa 0/0
Cisco_X(config-if)# ip address 192.77.203.1 255.255.255.0
Cisco_X(config-if)# no shutdown
Cisco_X(config-if)# exit
Cisco_X(config)# int se 0/0
Cisco_X(config-if)# clock rate 72000
Cisco_X(config-if)# bandwidth 72
Cisco_X(config-if)# ip address 200.100.100.1 255.255.255.252
Cisco_X(config-if)# no shutdown
Cisco_X(config-if)# exit
Cisco_X(config)# router ospf 100
Cisco_X(config-router)# network 200.100.100.0 0.0.0.255 area 0.0.0.0
Cisco_X(config-router)# passive-interface fa0/0
```

Cisco Y

```
Router(config)# hostname Cisco_Y
(config)# ip routing
Cisco_Y(config)# int fa 0/0
Cisco_Y(config-if)# ip address 192.168.1.1 255.255.255.0
Cisco_Y(config-if)# no shutdown
Cisco_Y(config-if)# exit
Cisco_Y(config)# int se 0/0
Cisco_Y(config-if)# clock rate 72000
Cisco_Y(config-if)# bandwidth 72
Cisco_Y(config-if)# ip address 200.100.100.10 255.255.255.252
Cisco_Y(config-if)# no shutdown
Cisco_Y(config-if)# exit
Cisco_Y(config)# router ospf 100
Cisco_Y(config-router)# network 200.100.100.0 0.0.0.255 area 0.0.0.0
Cisco_Y(config-router)# passive-interface fa0/0
```



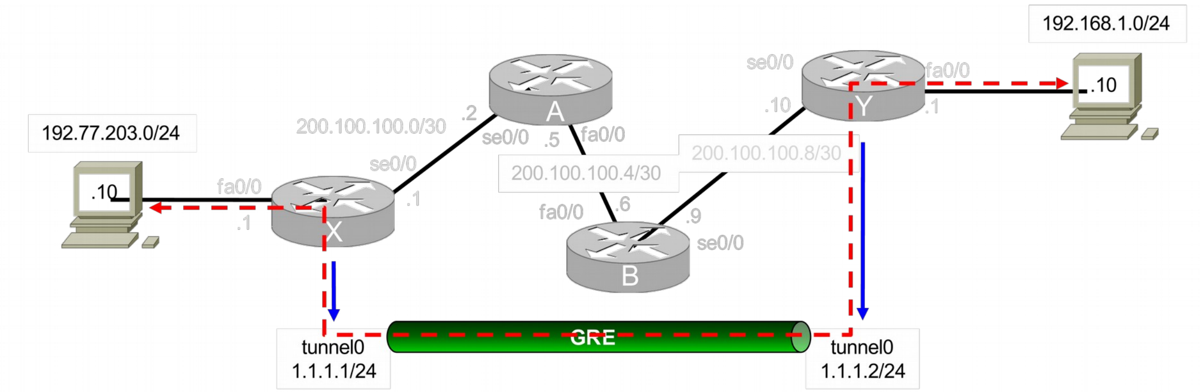
2.1.3 GRE tunnel configuration

Cisco X

```
Cisco_X(config)# int tunnel 0
Cisco_X(config-if)# ip address 1.1.1.1 255.255.255.0
Cisco_X(config-if)# tunnel source se0/0
Cisco_X(config-if)# tunnel destination 200.100.100.10
Cisco_X(config-if)# ip tcp adjust-mss 1436
Cisco_X(config-if)# tunnel key 1234
Cisco_X(config-if)# exit
Cisco_X(config)# ip route 192.168.1.0 255.255.255.0 1.1.1.2
```

Cisco Y

```
Cisco_Y(config)# int tunnel 0
Cisco_Y(config-if)# ip address 1.1.1.2 255.255.255.0
Cisco_Y(config-if)# tunnel source se0/0
Cisco_Y(config-if)# tunnel destination 200.100.100.1
Cisco_Y(config-if)# ip tcp adjust-mss 1436
Cisco_Y(config-if)# tunnel key 1234
Cisco_Y(config-if)# exit
Cisco_X(config)# ip route 192.77.203.0 255.255.255.0 1.1.1.1
```



Host X

```
Host_X# ping 192.168.1.10
Sending 5, 100-byte ICMP Echos to 192.168.1.10, timeout is 2 seconds.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/64 ms
Host_X#

Host_X# traceroute 192.168.1.10
Tracing the route to 192.168.1.10
1. 192.77.203.1 0 msec 0 msec 0 msec
2. 1.1.1.1 40 msec 36 msec 40 msec
3. 192.168.1.10 36 msec 36 msec *
Host_X#
```

Host Y

```
Host_Y# ping 192.77.203.10
Sending 5, 100-byte ICMP Echos to 192.77.203.10, timeout is 2 seconds.
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/64 ms
Host_Y#

Host_Y# traceroute 192.77.203.10
Tracing the route to 192.77.203.10
1. 192.168.1.1 0 msec 0 msec 0 msec
2. 1.1.1.1 40 msec 36 msec 40 msec
3. 192.77.203.10 36 msec 36 msec *
Host_Y#
```

2.1.4 Testing the GRE tunnel configuration

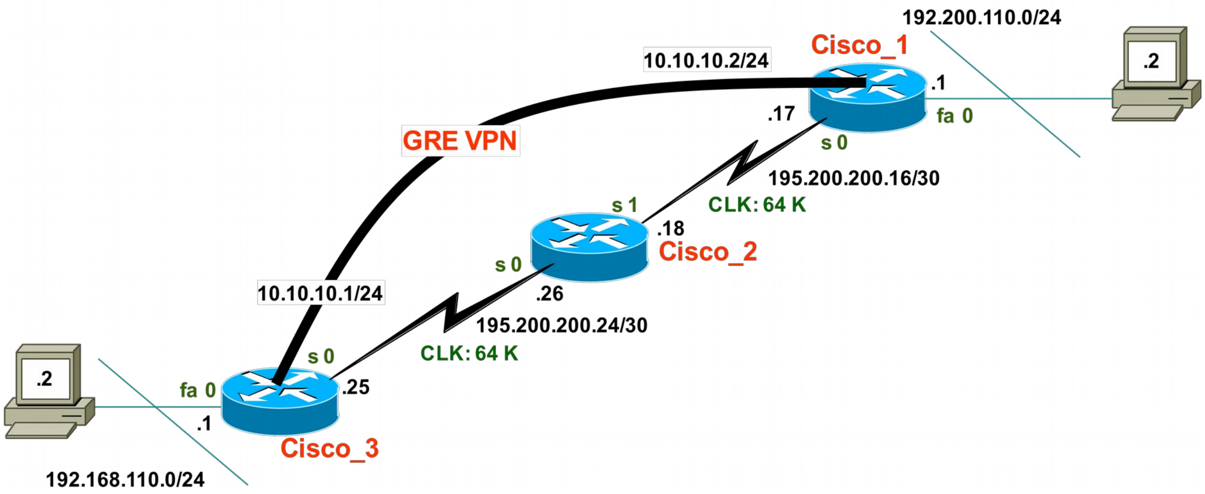
Host X

```
Host_X# ping 192.168.1.10
Sending 5, 100-byte ICMP Echos to 192.168.1.10, timeout is 2 seconds.
! ! ! ! !
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/64 ms
Host_X#
Host_X# traceroute 192.168.1.10
Tracing the route to 192.168.1.10
1. 192.77.203.1 0 msec 0 msec 0 msec
2. 1.1.1.1 40 msec 36 msec 40 msec
3. 192.168.1.10 36 msec 36 msec *
Host_X#
```

Host Y

```
Host_Y# ping 192.77.203.10
Sending 5, 100-byte ICMP Echos to 192.77.203.10, timeout is 2 seconds.
! ! ! ! !
Success rate is 100 percent (5/5), round-trip min/avg/max = 64/64/64 ms
Host_Y#
Host_Y# traceroute 192.77.203.10
Tracing the route to 192.77.203.10
1. 192.168.1.1 0 msec 0 msec 0 msec
2. 1.1.1.1 40 msec 36 msec 40 msec
3. 192.77.203.10 36 msec 36 msec *
Host_Y#
```

3 Lab Exercise - Configuring GRE



3.1 Objective

- Practice building GRE VPN Links.
- Practice testing GRE VPN Links.

3.2 Background

Knowing how to configure routers for interconnecting LANs via GRE VPN is an essential building block to you networking knowledge.

3.3 Lab Steps

Physically build network as shown.

4 Internet Protocol Security (IPsec)

IPsec is a suite of protocols for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a data stream. IPsec also includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session. IPsec can be used to protect data flows between a pair of hosts (e.g. computer users or servers), between a pair of security gateways (e.g. routers or firewalls), or between a security gateway and a host.

IPsec is an end-to-end security solution and operates at the Internet Layer of the Internet Protocol Suite, comparable to Layer 3 in the OSI model. Other Internet security protocols in widespread use, such as SSL, TLS and SSH, operate in the upper layers of these models. This makes IPsec more flexible, as it can be used for protecting all the higher level protocols, because applications don't need to be designed to use IPsec, whereas the use of TLS/SSL or other higher-layer protocols must be incorporated into the design of an application.

IPsec is the successor of the NLSP (Network Layer Security Protocol) that was standardised by ISO. The NLSP protocol was directly taken from the SP3 protocol that was published by NIST, but defined by the Secure Data Network System project of the NSA.

The term "IPsec" is officially defined by the Internet Engineering Task Force (IETF). This definition includes the form of capitalization used for the term; it is often incorrectly spelled IPsec.

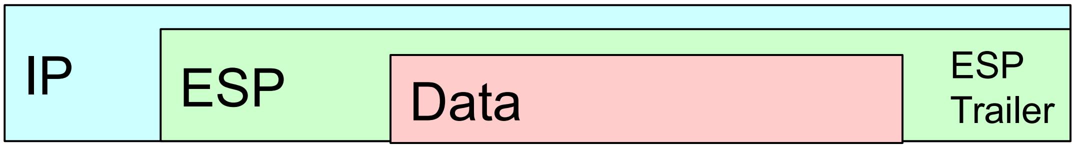
IPsec integrates security directly into IP. IPsec provides three main areas of security:

- Authentication - which validates the communicating parties;
- Integrity - which makes sure the data has not been altered; and
- Confidentiality - which ensures the data cannot be intercepted and viewed.

IPsec secures the underlying network layer. That way, an IPsec link is secure regardless of the application. IPsec works with the existing Internet infrastructure using encapsulation. It secures a packet of data by encrypting it before sending it over the Internet. On the receiving end, an IPsec-compliant device decrypts the data.

The security protection can be selectively applied to various types of data traffic based on protocols, IP addresses, network addresses, applications (via TCP/UDP port addresses), and network interfaces. System-originated IP traffic (Telnet, OSPF, and RIP for example) can be protected by IPsec directly. Other multiprotocol traffic (IPX, AppleTalk, and DECnet for example) and forwarded IP traffic are protected by IPsec through the L2TP/PPTP tunnel.

4.1 Encapsulation security payload (ESP)



Encapsulation security payload (ESP) is defined in RFC 2407 and is used to provide data confidentiality via encryption. For outbound traffic, it encrypts the IP payload and inserts an ESP header between the IP header and the payload. For inbound traffic, it decrypts the IP payload and removes the ESP header.

4.1.1.1 Encapsulating Security Payload Fields

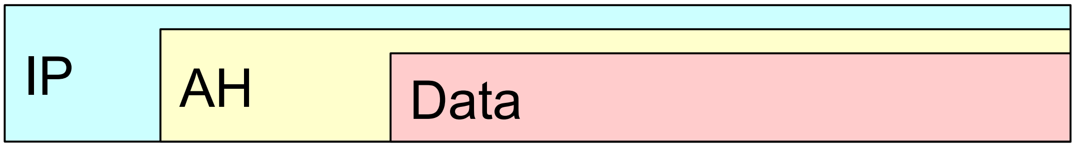
ESP has several fields that are the same as those used in AH, but packages its fields in a very different way. Instead of having just a header, it divides its fields into three components:

ESP Header: This contains two fields, the SPI and Sequence Number, and comes before the encrypted data. Its placement depends on whether ESP is used in transport mode or tunnel mode, as explained in the topic on IPSec modes.

ESP Trailer: This section is placed after the encrypted data. It contains padding that is used to align the encrypted data, through a Padding and Pad Length field. Interestingly, it also contains the Next Header field for ESP.

ESP Authentication Data: This field contains an Integrity Check Value (ICV), computed in a manner similar to how the AH protocol works, for when ESP's optional authentication feature is used.

4.2 Authentication header (AH)



AH is defined by RFC 2402 and is used to provide data integrity and data origin authentication and to provide protection against replays using the MD5-HMAC or SHA1-HMAC crypto algorithm. For outbound traffic, AH computes integrity checksum value (ICV) and inserts an authentication header between the IP header and the higher layer protocol header. For inbound traffic, AH verifies the ICV and removes the AH. AH can be applied alone or with ESP.

MD5-HMAC and SHA1-HMAC are standards-based hash algorithms. In general, SHA1-HMAC requires more computation and is considered to be more secure but slower.

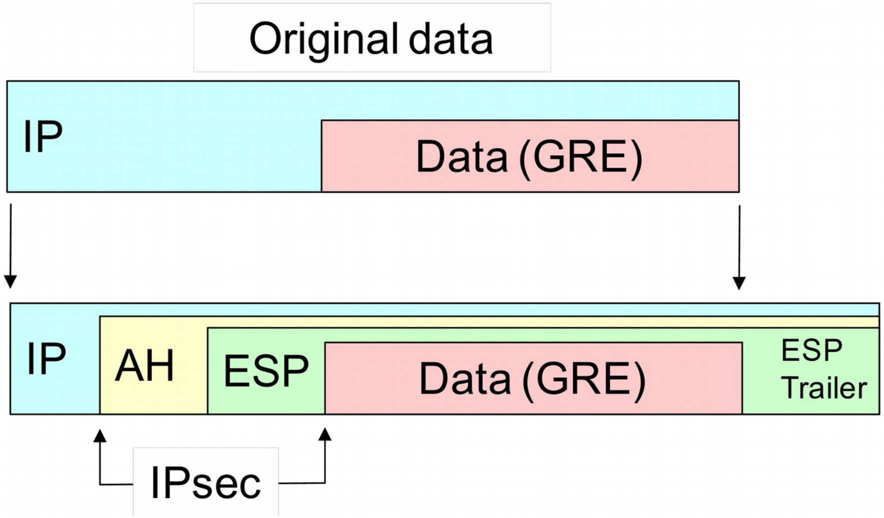
- SHA1 – Secure Hash Algorithm 1
- MD5 – Message Digest 5

IPsec Modes

- Transport Mode
- Tunnel Mode

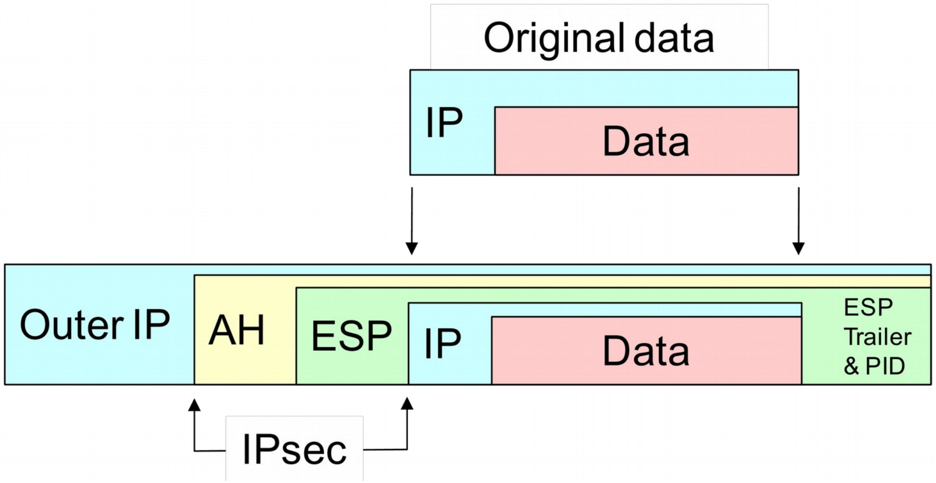
Manual or Dynamic Key

- Dynamic key
- Manual Key



4.2.1 IPsec Encryption Transport mode

Transport mode security associations are used to protect traffic that is viewed on an end system from an IPsec perspective. For example, it can be used with GRE/PPTP/L2TP tunnels or to serve network management traffic like Telnet or SNMP. In effect Transport mode encrypts the whole tunnel passing from one router to another.

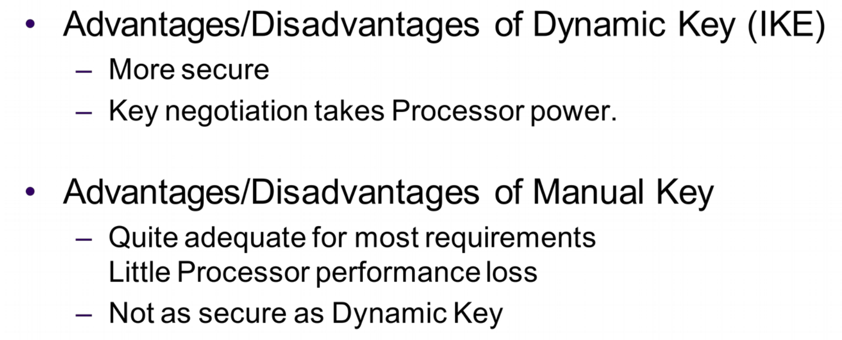


4.2.2 IPsec Encryption Tunnel mode

Tunnel mode security associations are used to protect IP traffic forwarded by the router on IPsec tunnel ports. When IPsec-compliant encryption is applied to an entire network protocol packet (IP, IPX, AppleTalk, etc.), and then the encrypted results are encapsulated into another IP packet, the process is called "tunnelling mode."

The advantage of using this mode is that a network protocol can travel across a network that does not support it to a tunnel termination device that does. Tunnelling mode also protects the identity of networks, subnetworks, and terminating nodes. To confuse the picture further, Layer 2 VPNs provide these same benefits, whether or not they incorporate IPsec.

4.3 IPsec Keys



- Advantages/Disadvantages of Dynamic Key (IKE)
  - More secure
  - Key negotiation takes Processor power.
- Advantages/Disadvantages of Manual Key
  - Quite adequate for most requirements
    Little Processor performance loss
  - Not as secure as Dynamic Key

4.3.1 Manual Key

The key is added to the policy file. When the key is entered no particular length restriction is applied. Keys can be entered as either ASCII text or hex values in the range of 1 to 128 bytes. When a key is bound, certain length restrictions are applied.

A Security Parameters Index (SPI) value is used in conjunction with the destination address to identify a particular security association which represents a set of agreements between senders and receivers on a key, on an encryption or authentication algorithm, and on SPI numbers.

4.3.2 Dynamic key

This method used Internet Key Exchange (IKE) alongside IPsec. The IKE protocol is a key management protocol standard defined in RFC 2409. It incorporates the Oakley key exchange and Skeme key exchange inside the Internet Security Association and Key Management Protocol (ISAKMP) framework. It is therefore a hybrid protocol, which can be used to negotiate, and provide authenticated keying material for, security associations in a protected manner.

IKE, when used in conjunction with IPsec provides for Dynamic Key Exchange and additional flexibility. It thus improves the scalability of IPsec over the broad spectrum of large network deployments over publicly shared infrastructures like the Internet. IKE uses either Lifetime or Perfect Forward Secrecy (PFS) system to initiate and control key negotiation.

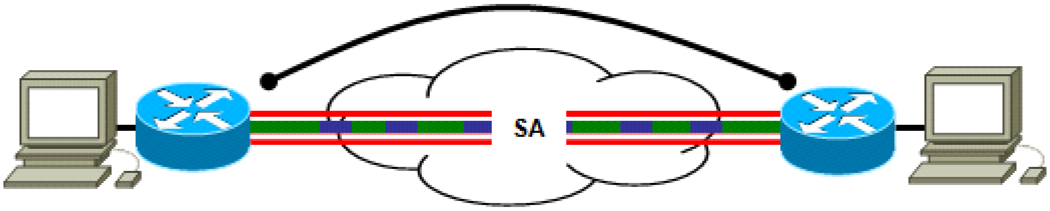
Lifetime determines the amount of time elapsed and/or the amount of data protected by an IPsec security association before it expires. The lifetime can be specified in units of minute's (m), hours (h), days(d), and/or kilobytes (kb), and megabytes (mb).

PFS provides higher security by renegotiating a shared secret between IPsec peers each time a new key is needed. Since generating a shared secret demands intense numerical calculations (know as Diffie-Hellman), using this option may cause reduced performance during renegotiations.

4.4 Internet Key Exchange (IKE)

Internet Key Exchange (IKE or IKEv2) is the protocol used to set up a security association (SA) in the IPsec protocol suite. IKE uses a Diffie-Hellman key exchange to set up a shared session secret, from which cryptographic keys are derived. Public key techniques or, alternatively, a pre-shared key, are used to mutually authenticate the communicating parties. IKE builds upon the Oakley protocol.

4.5 Internet Security Association & Key Management



ISAKMP defines the procedures for authenticating a communicating peer, creation and management of Security Associations, key generation techniques, and threat mitigation (e.g. denial of service and replay attacks). ISAKMP typically utilises IKE for key exchange, although other methods can be implemented. Preliminary SA is formed using this protocol; later a fresh keying is done.

ISAKMP defines procedures and packet formats to establish, negotiate, modify and delete Security Associations. SAs contain all the information required for execution of various network security services, such as the IP layer services (such as header authentication and payload encapsulation), transport or application layer services, or self-protection of negotiation traffic. ISAKMP defines payloads for exchanging key generation and authentication data. These formats provide a consistent framework for transferring key and authentication data which is independent of the key generation technique, encryption algorithm and authentication mechanism.

ISAKMP is distinct from key exchange protocols in order to cleanly separate the details of security association management (and key management) from the details of key exchange. There may be many different key exchange protocols, each with different security properties. However, a common framework is required for agreeing to the format of SA attributes, and for negotiating, modifying, and deleting SAs. ISAKMP serves as this common framework.

ISAKMP can be implemented over any transport protocol. All implementations must include send and receive capability for ISAKMP using UDP on port 500. Additionally, UDP port 4500 must also be allowed at the destination if the source interface IP address undergoes network address translation from natural (assigned) IP address to a public IP address for connection to the internet.

- ISAKMP
  - Is a framework used by IKE
  - Carries Key Management Information.
  - UDP based (port 500).

- Authenticate and Establish Secure Channel (SA).
- Exchange Encryption Key over Secure Channel and Parameter Negotiation (SA).
- Tunnel Traffic Starts to Flow (IPsec - ESP).
- Periodic Encryption Key Renewals (SA).

4.5.1 OAKLEY

OAKLEY is the key determination protocol currently mandated for compliant implementations by the IETF. The OAKLEY protocol is related to the Station-To-Station (STS) protocol. It shares the similarity of authenticating the Diffie-Hellman exponent exchange and their subsequent use in computing a shared key with STS.

The authentication of the DH exponentials is required due to the vulnerability of the basic Diffie-Hellman mechanism to man-in-the-middle attacks. OAKLEY uses public key cryptography or out of band, pre-shared symmetric keys to authenticate the participants in the DH exponential exchange.

IKE is a protocol that describes methods to obtain authenticated keying material for use with ISAKMP and the Security protocols, for the IETF IP Domain of Interpretation. It is a hybrid protocol based on ISAKMP, OAKLEY and SKEME.

A typical ISAKMP/OAKLEY exchange is divided into two phases. The first phase establishes a security association and a session key between the ISAKMP peers. The second phase establishes SAs and session keys for the security protocols (AH and ESP).

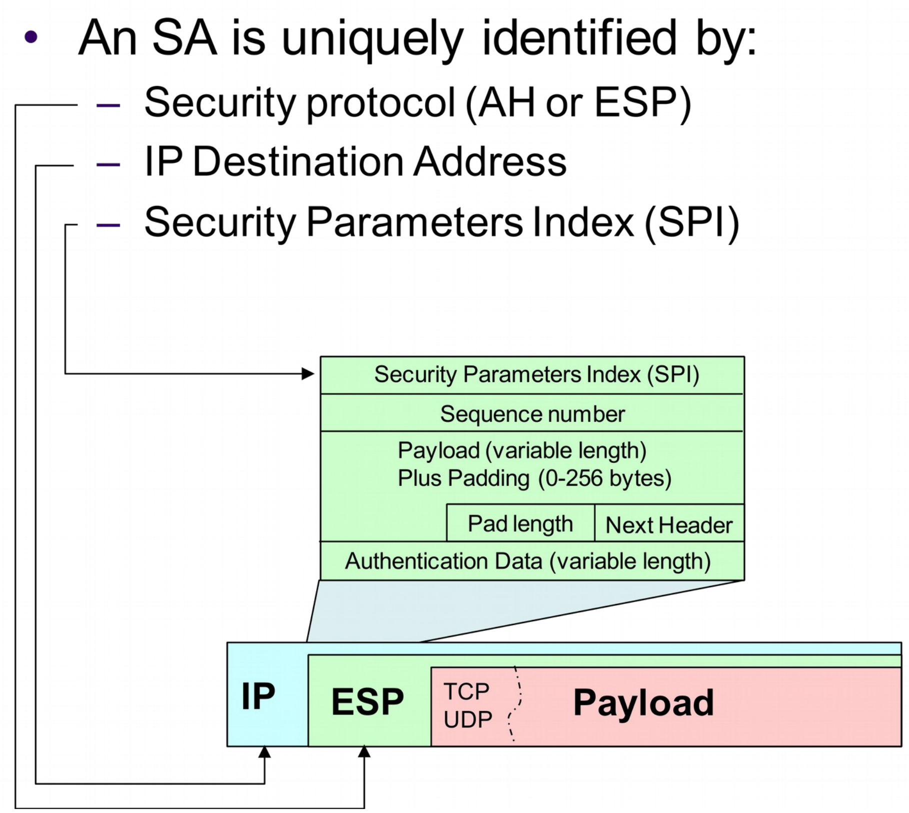
4.5.2 IKE Features

Main Features of IKE are:

- **Shared Key Confidentiality**
  - Ensures privacy of the key used for negotiation phases.
- **Encryption Key Confidentiality**
  - Ensures the privacy of the key used to encrypt data.
- **Authentication**
  - Ensures that the initiator and the responder are who they claim to be. Identity protection is one of the benefits of IKE.
- **Other features**
  - Allows varying degree of strength in key exchange security.
  - Lifetime of Security Associations ensure added security for highly sensitive data.

4.6 Security Association ID

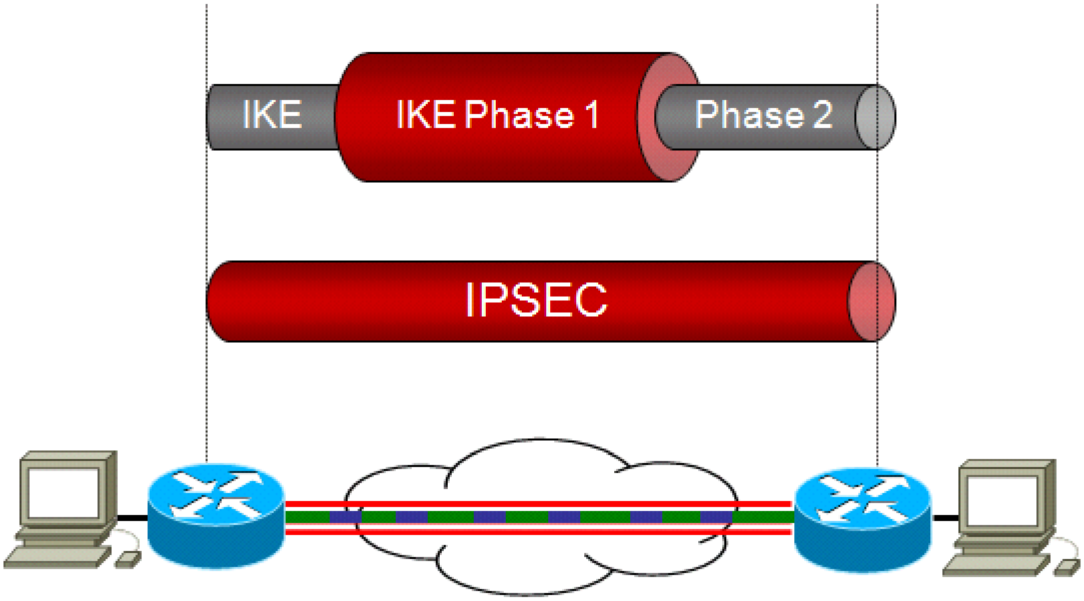
The concept of a security association (SA) is fundamental to IPsec. An SA is a relationship between two or more entities that describes how the entities will use security services to communicate securely. IPsec provides many options for performing network encryption and authentication. Each IPsec connection can provide encryption, integrity, authenticity, or all three services. When the security service is determined, the two IPsec peers must determine exactly which algorithms to use (for example, DES or 3DES for encryption; MD5 or SHA-1 for integrity). After deciding on the algorithms, the two devices must share session keys. As you can see, there is quite a bit of information to manage. The security association is the method that IPsec uses to track all the particulars concerning a given IPsec communication session.



- An SA is uniquely identified by:
  - Security protocol (AH or ESP)
  - IP Destination Address
  - Security Parameters Index (SPI)

Security Parameters Index (SPI)
Sequence number
Payload (variable length)
Plus Padding (0-256 bytes)
Pad length
Next Header
Authentication Data (variable length)
IP
ESP
TCP
UDP
Payload

4.7 IKE Negotiation



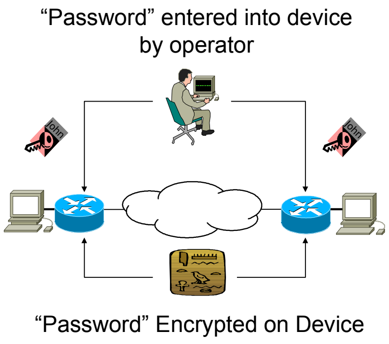
4.7.1 IKE Primary Authentication

IKE must authenticate the identities of the systems using the Diffie-Hellman algorithm. This process is known as primary authentication. IPsec IKE can use two primary authentication methods:

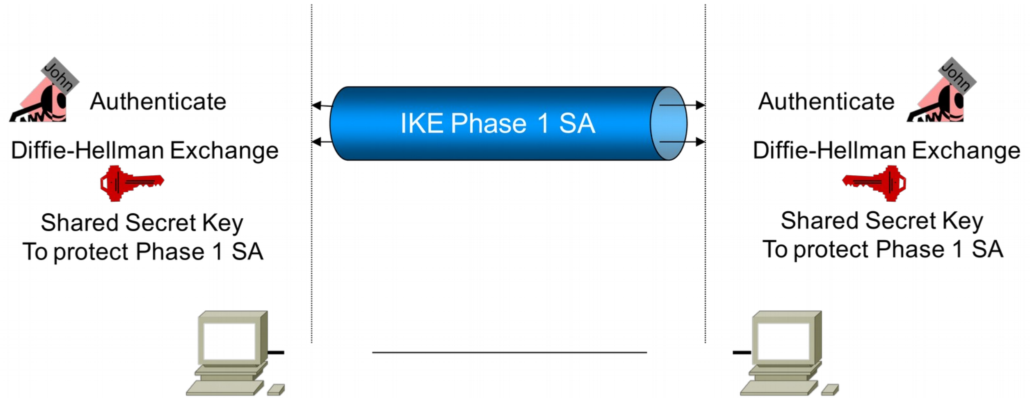
- Digital Signatures
- Pre-shared keys

Digital signature and public-key encryption are both based on asymmetric key encryption and require a mechanism for distributing public keys. This is usually done using security certificates and a Public Key Infrastructure (PKI).

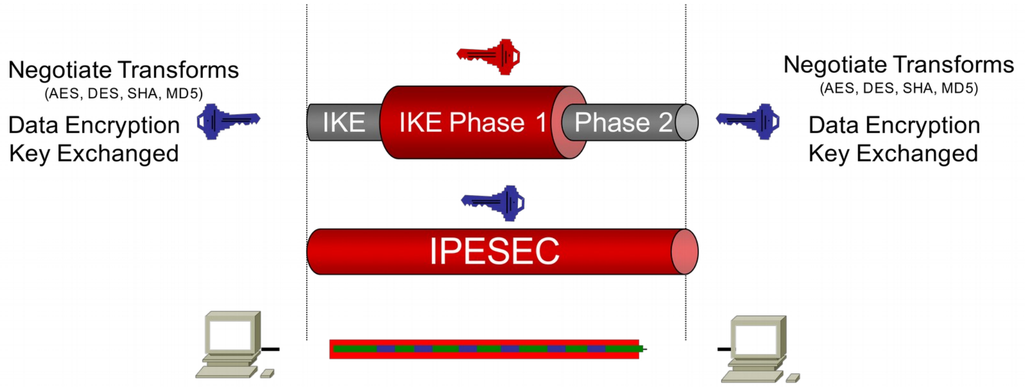
4.7.2 IKE Pre-shared Key


"Password" entered into device by operator
John
John
"Password" Encrypted on Device

With pre-shared key authentication, the two entities must manually exchange and configure a shared, symmetric key. Note that the pre-shared key is used only for the primary authentication. The two negotiating entities then generate dynamic shared keys for the IKE SAs. Pre-shared keys do not require a Certificate Authority or Public Key Infrastructure.

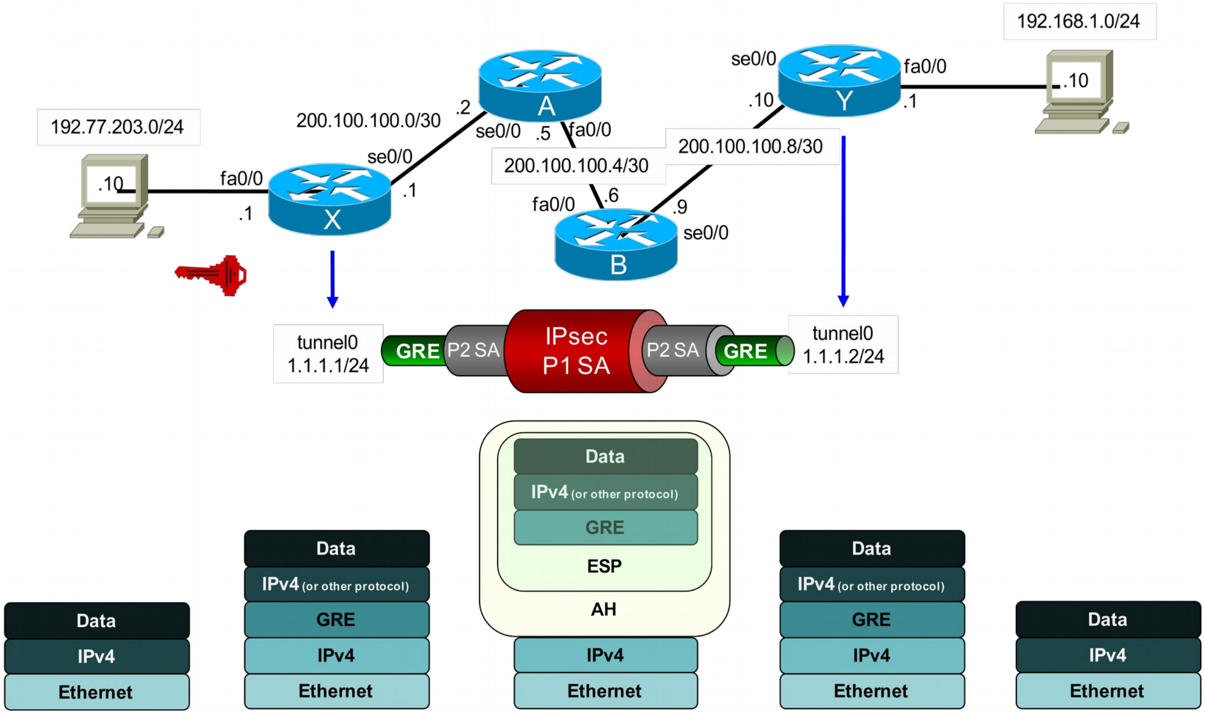


Authentication pass before any key exchange takes place.



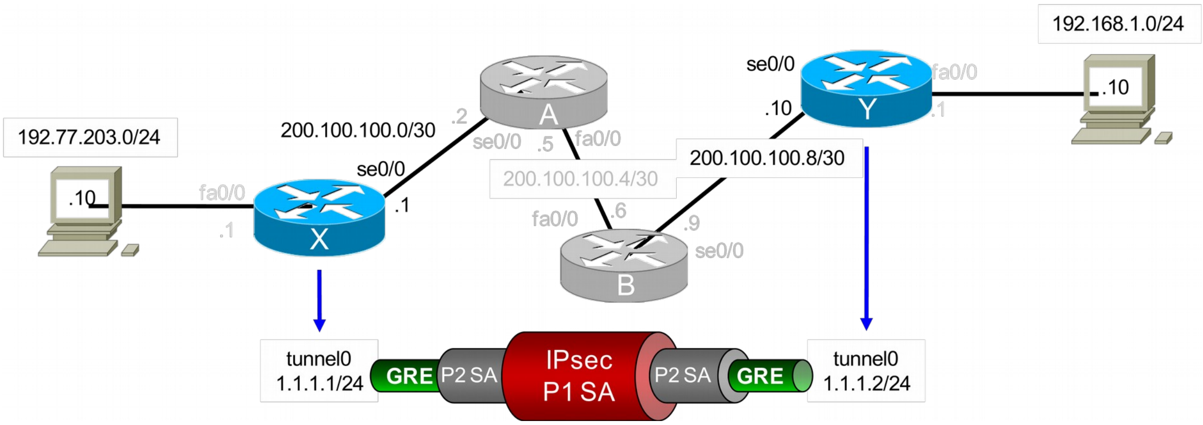
Data Encryption Key can be refreshed through Phase 2 SA.

5 Configuring the GRE/IPsec Tunnel



In the diagram we can see that the configuration of the GRE tunnel remains as is. We will add IPsec to all packets that are the GRE tunnel. This has disadvantages in terms of multiple layers of encapsulation but has the advantage that multicast traffic can be carried transparently within the GRE tunnel as it is the tunnel that is encapsulated by IPsec and not the traffic within it. A point to point IPsec tunnel can be created however it is the IP packets that are directly encapsulated and therefore such a tunnel cannot handle IP multicast.

You will see from the model that the traffic is firstly encapsulated within GRE and then the GRE packet data (the GRE tunnel) is encapsulated within the AH and ESP headers of IPsec.



5.1 The ISAKMP Policy

An Internet Key Exchange (IKE) policy is created which is identical at both sides of the link. This policy defines a set of parameters to be used during the IKE negotiation of Phase 1 Secure Association. The policy is assigned a priority number, in this case 1 which is used to uniquely identify the IKE policy and assigns a priority to the policy.

```
Cisco_X(config)# crypto isakmp policy 1              (1 – 10000 with 1 as the highest)
Cisco_X(config-isakmp)# encryption aes               (AES / DES)
Cisco_X(config-isakmp)# hash sha                     (MD5 / SHA1)
Cisco_X(config-isakmp)# authentication pre-share
Cisco_X(config-isakmp)# lifetime 86400               (86,400 seconds or one day)
Cisco_X(config-isakmp)# group 2                      (1 - 768-bit, 2 – 1024-bit, 5 – 1536-bit  Diffie-Hellman)
Cisco_X(config-isakmp)# exit

Cisco_Y(config)# crypto isakmp policy 1              (1 – 10000 with 1 as the highest)
Cisco_Y(config-isakmp)# encryption aes               (AES / DES)
Cisco_Y(config-isakmp)# hash sha                     (MD5 / SHA1)
Cisco_Y(config-isakmp)# authentication pre-share
Cisco_Y(config-isakmp)# lifetime 86400               (86,400 seconds or one day)
Cisco_Y(config-isakmp)# group 2                      (1 - 768-bit, 2 – 1024-bit, 5 – 1536-bit  Diffie-Hellman)
Cisco_Y(config-isakmp)# exit
```

Once this is complete on both sides of the link we must define the identity used by the router when participating in the Internet Key Exchange (IKE) protocol. During Phase I ISAKMP negotiations the peers must identify themselves to each other. This can be by the IP Address, Distinguished name or the router Hostname. Here we use the IP addresses of the hosts exchanging ISAKMP identity information as the identification method. We also define the pre-shared authentication key and the peer IP address of remote peer.

```
Cisco_X(config)# crypto isakmp identity address
Cisco_X(config)# crypto isakmp key ipsec_secret address 200.100.100.10

Cisco_Y(config)# crypto isakmp identity address
Cisco_Y(config)# crypto isakmp key ipsec_secret address 200.100.100.1
```

We must now define a transform set on each side. A transform set is an acceptable combination of security protocols, algorithms, and other settings to apply to IPSec protected traffic. During the IPSec security association (SA) negotiation, the peers agree to use a particular transform set when protecting a particular data flow. You can configure multiple transform sets, and then specify one or more of these transform sets in a crypto map entry.

```
Cisco_X(config)# crypto ipsec transform-set ts1 ah-sha-hmac esp-aes
Cisco_X(cfg-ctypto-trans)# mode transport                         (transport | tunnel)
Cisco_X(cfg-ctypto-trans)# exit

Cisco_Y(config)# crypto ipsec transform-set ts1 ah-sha-hmac esp-aes
Cisco_Y(cfg-ctypto-trans)# mode transport
Cisco_Y(cfg-ctypto-trans)# exit
```

We must now create a crypto map on each side. The crypto maps provide two functions: filtering and classifying traffic to be protected and defining the policy to be applied to that traffic. The first use affects the flow of traffic on an interface; the second affects the negotiation performed (via IKE) on behalf of that traffic.

IPSec crypto maps define the following:

- What traffic should be protected (via and access-list filter).
- To which IPsec peer the protected traffic can be forwarded, this is the remote peer with which an SA can be established.
- Which transform sets are acceptable for use with the protected traffic.
- How keys and security associations should be used or managed (or what the keys are, if IKE is not used).

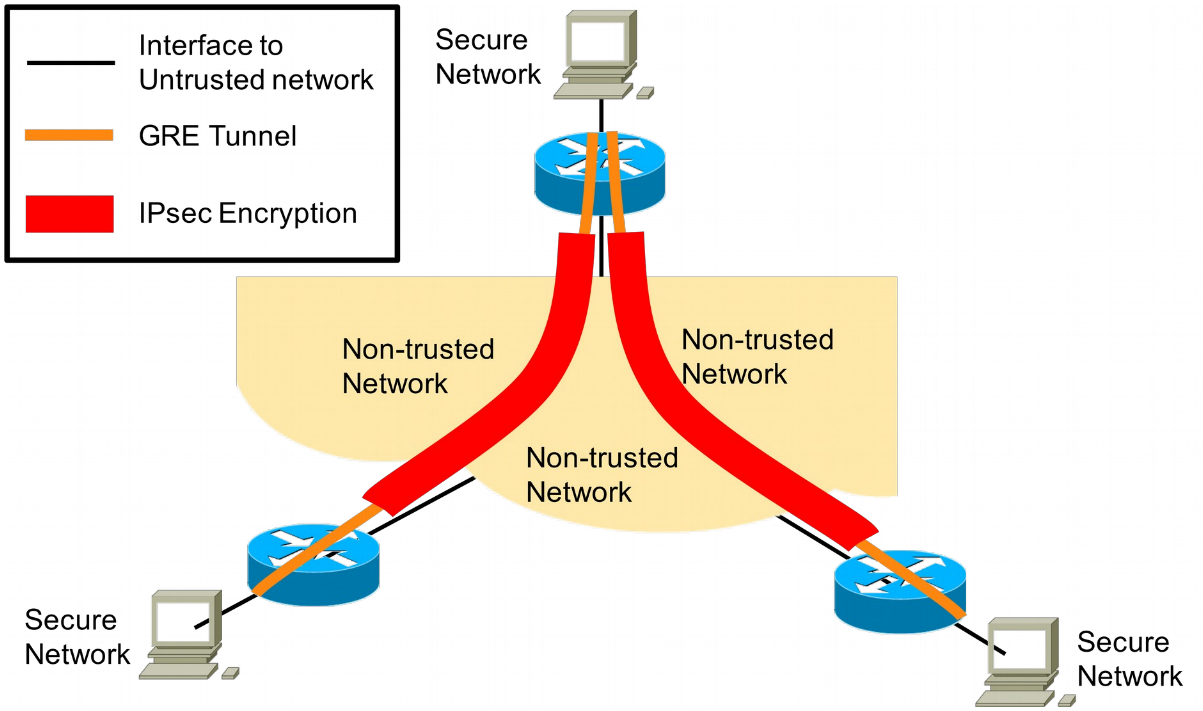
```
Cisco_X(config)# access-list 110 permit gre host 200.100.100.1 host 200.100.100.10

Cisco_X(config)# crypto map vpn 10 ipsec-isakmp            (ipsec-isakmp | ipsec-manual | dynamic)
Cisco_X(cfg-ctypto-map)# match address 110                 (110 is the access-list number)
Cisco_X(cfg-ctypto-map)# set peer 200.100.100.10
Cisco_X(cfg-ctypto-map)# set transform-set ts1             (as defined earlier i.e. ts1)
Cisco_X(cfg-ctypto-map)# exit

Cisco_Y(config)# access-list 110 permit gre host 200.100.100.10 host 200.100.100.1

Cisco_Y(config)# crypto map vpn 10 ipsec-isakmp
Cisco_Y(cfg-ctypto-map)# match address 110
Cisco_Y(cfg-ctypto-map)# set peer 200.100.100.1
Cisco_Y(cfg-ctypto-map)# set transform-set ts1
Cisco_Y(cfg-ctypto-map)# exit
```

We must now assign the crypto map set to both the physical public interface and the tunnel. You must assign a crypto map set to an interface before that interface can provide IPSec services. Only one crypto map set can be assigned to an interface.

Note: A crypto map cannot be applied to a loopback interface.

```
Cisco_X(config)# interface Tunnel0
Cisco_X(config-if)# crypto map vpn
Cisco_X(config-if)# exit
Cisco_X(config)# interface se 0/0
Cisco_X(config-if)# crypto map vpn
Cisco_X(config-if)# exit

Cisco_Y(config)# interface Tunnel0
Cisco_Y(config-if)# crypto map vpn
Cisco_Y(config-if)# exit
Cisco_Y(config)# interface se 0/0
Cisco_Y(config-if)# crypto map vpn
Cisco_Y(config-if)# exit
```

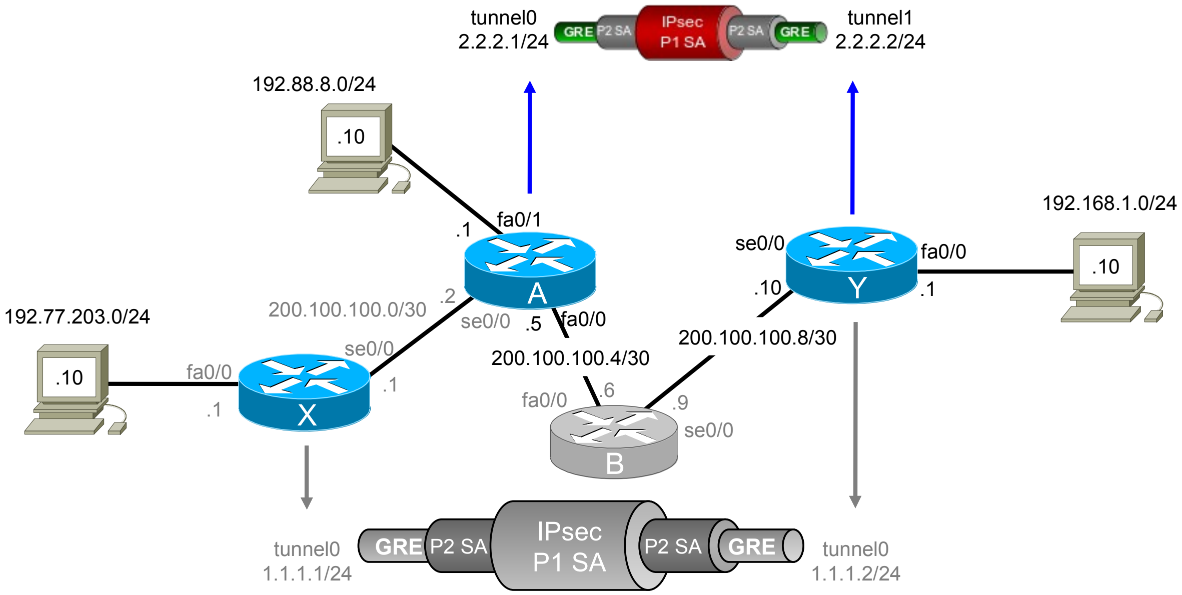
5.2 GRE/IPSec 3-way Tunnel



Thus far we have looked at single point to point networks. What if we wanted to have a secure point to multipoint type network overlaid upon a non trusted network.

Well we can build a mesh or partial mesh of GRE tunnels which will form the basis of the overlay network. We will then encrypt the GRE traffic between these points as in the diagram above.

5.2.1 Building the additional tunnel and encryption



Cisco Y

This router becomes the convergence point for the existing tunnel and the new tunnel to Cisco_A.

The new tunnel is created pointing to Cisco_A's fa0/0 interface.

```
Cisco_Y(config)# int tunnel 1
Cisco_Y(config-if)# ip address 2.2.2.2 255.255.255.0
Cisco_Y(config-if)# tunnel source se0/0
Cisco_Y(config-if)# tunnel destination 200.100.100.5
Cisco_Y(config-if)# ip tcp adjust-mss 1436
Cisco_Y(config-if)# tunnel key 5678
Cisco_Y(config)# exit
```

A route to the new network on Cisco_A fa0/1 interface is configured. (Remember as this is the trusted network we do not want OSPF to advertise these).

```
Cisco_Y(config)# ip route 192.88.8.0 255.255.255.0 2.2.2.1
```

Access list to define this tunnel for IPsec.

```
Cisco_Y(config)# access-list 111 permit gre host 200.100.100.10 host 200.100.100.5
```

Create shared ISAKMP Key for phase I with Cisco_A.

```
Cisco_Y(config)# crypto isakmp key 0 ipsec_2secret address 200.100.100.5
```

Create additional crypto map sequence for the crypto map 'vpn'. It is possible to create a second crypto map however only one crypto map can be associated with each interface and as in this configuration the intention is to use se0/0 for both tunnels we cannot use a second crypto map.

```
Cisco_Y(config)# crypto map vpn 11 ipsec-isakmp
Cisco_Y(cfg-ctypto-map)# match address 111
Cisco_Y(cfg-ctypto-map)# set peer 200.100.100.5
Cisco_Y(cfg-ctypto-map)# set transform-set ts1
Cisco_Y(cfg-ctypto-map)# exit
```

Apply the crypto map 'vpn' to the new tunnel. It is already applied to se0/0 from the earlier configuration of the point to point link.

```
Cisco_Y(config)# interface Tunnel1
Cisco_Y(config-if)# crypto map vpn
Cisco_Y(config-if)# exit
```

Cisco A

This router becomes the convergence point for the existing tunnel and the new tunnel to Cisco_A.

Add the additional Ethernet network 192.88.8.0/24 on fa0/1.

```
Cisco_A(config)# interface FastEthernet0/0
Cisco_A(config-if)# ip address 192.88.8.0 255.255.255.0
Cisco_A(config-if)# no shutdown
Cisco_A(config-if)# exit
```

Create a GRE tunnel to connect to tunnel1 on Cisco_Y.

```
Cisco_A(config)# int tunnel 0
Cisco_A(config-if)# ip address 2.2.2.1 255.255.255.0
Cisco_A(config-if)# tunnel source fa0/0
Cisco_A(config-if)# tunnel destination 200.100.100.10
Cisco_A(config-if)# ip tcp adjust-mss 1436
Cisco_A(config-if)# tunnel key 5678
Cisco_A(config)# exit
```

Create IP routes to the two trusted networks. Remember we do not want these routes advertised by OSPF as it is controlling the routes of the non-trusted network.

```
Cisco_A(config)# ip route 192.168.1.0 255.255.255.0 2.2.2.2
Cisco_A(config)# ip route 192.77.203.0 255.255.255.0 2.2.2.2
```

Create an ISAKMP policy.

```
Cisco_A(config)# crypto isakmp policy 1
Cisco_A(config-isakmp)# encryption aes
Cisco_A(config-isakmp)# hash sha
Cisco_A(config-isakmp)# authentication pre-share
Cisco_A(config-isakmp)# lifetime 86400
Cisco_A(config-isakmp)# group 2
Cisco_A(config-isakmp)# exit
```

During Phase I ISAKMP negotiations the peers must identify themselves to each other. This uses the IP addresses of the hosts exchanging ISAKMP identity information as the identification method.

```
Cisco_A(config)# crypto isakmp identity address
```

Create shared ISAKMP Key for phase I with Cisco_Y.

```
Cisco_A(config)# crypto isakmp key 0 ipsec_2secret address 200.100.100.10
```

Create the Transform Set with the transport mode.

```
Cisco_A(config)# crypto ipsec transform-set ts1 ah-sha-hmac esp-aes
Cisco_A(cfg-ctypto-trans)# mode transport
Cisco_A(cfg-ctypto-trans)# exit
```

Access List to map the GRE tunnel traffic to the IPsec encryption.

```
Cisco_A(config)# access-list 110 permit gre host 200.100.100.5 host 200.100.100.10
```

Crypto map of the GRE tunnel identified in the Access List to IPSec to Cisco_Y as a peer.

```
Cisco_A(config)# crypto map vpn 10 ipsec-isakmp
Cisco_A(cfg-ctypto-map)# match address 110
Cisco_A(cfg-ctypto-map)# set peer 200.100.100.10
Cisco_A(cfg-ctypto-map)# set transform-set ts1
Cisco_A(cfg-ctypto-map)# exit
```

Assignment of the Crypto Map to the tunnel and ‘outer’ fa0/0 interface on Cisco_A. In this case fa0/1 is like a WAN interface connecting Cisco_A to the non-trusted network.

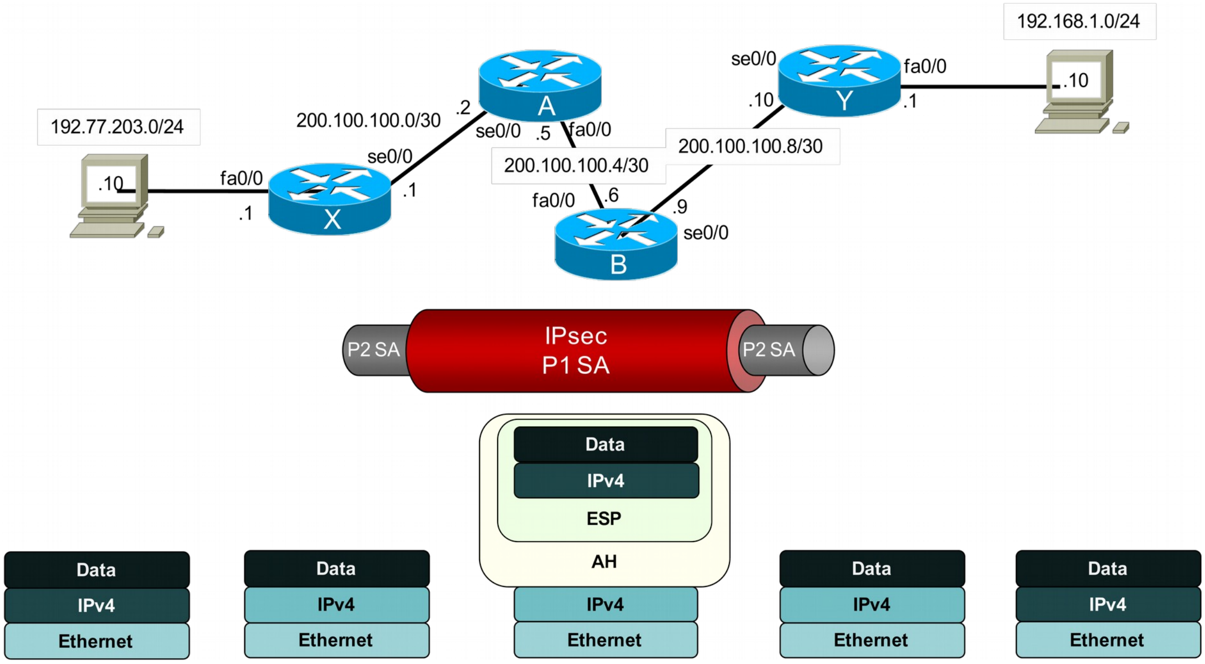
```
Cisco_A(config)# interface Tunnel0
Cisco_A(config-if)# crypto map vpn
Cisco_A(config-if)# exit
Cisco_A(config)# interface fa 0/0
Cisco_A(config-if)# crypto map vpn
Cisco_A(config-if)# exit
```

Cisco X

Add a route to the new trusted network on Cisco_A.

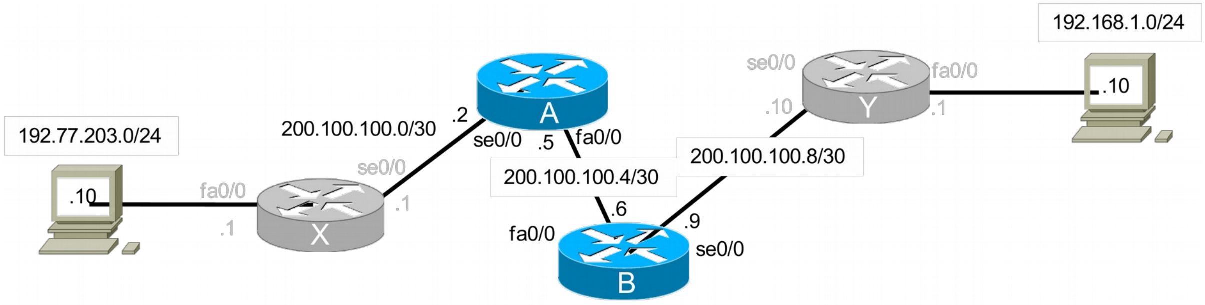
```
Cisco_X(config)# ip route 192.88.8.0 255.255.255.0 1.1.1.2
```

6 IPsec in Tunnel mode



The GRE tunnel is not essential and traffic can be carried through untrusted networks using IPsec tunnel mode. IPSec in tunnel mode can provide security for IP traffic only. The tunnel is configured to protect traffic between either two IP addresses or two IP subnets. If the tunnel is used between two computers instead of two gateways, the IP address outside the AH or ESP payload is the same as the IP address inside the AH or ESP payload. In Windows XP and the Windows Server 2003 family, IPSec does not support protocol-specific or port-specific tunnels.

6.1 Back to Core Router Configuration



Cisco A

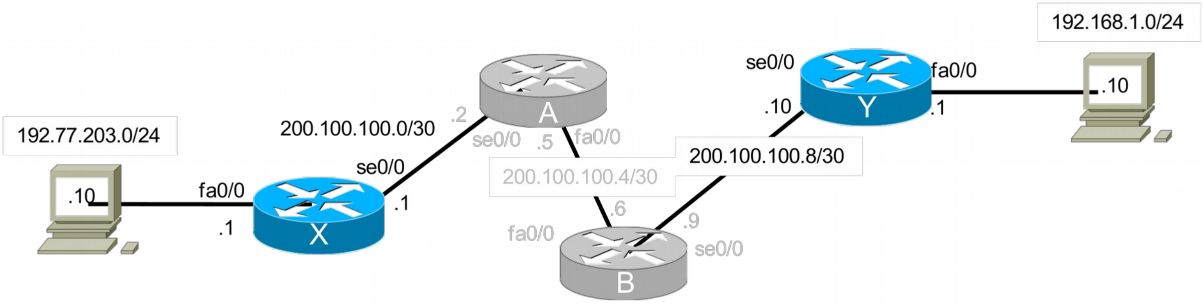
```
Router(config)# hostname Cisco_A
Cisco_A(config)# ip routing
Cisco_A(config)# int fa 0/0
Cisco_A(config-if)# ip address 200.100.100.5 255.255.255.252
Cisco_A(config-if)# no shutdown
Cisco_A(config-if)# exit
Cisco_A(config)# int se 0/0
Cisco_A(config-if)# clock rate 72000
Cisco_A(config-if)# bandwidth 72
Cisco_A(config-if)# ip address 200.100.100.2 255.255.255.252
Cisco_A(config-if)# no shutdown
Cisco_A(config-if)# exit
Cisco_A(config)# router ospf 100
Cisco_A(config-router)# network 200.100.100.0 0.0.0.255 area 0.0.0.0
```

Cisco B

```
Router(config)# hostname Cisco_B
Cisco_B(config)# ip routing
Cisco_B(config)# int fa 0/0
Cisco_B(config-if)# ip address 200.100.100.6 255.255.255.252
Cisco_B(config-if)# no shutdown
Cisco_B(config-if)# exit
Cisco_B(config)# int se 0/0
Cisco_B(config-if)# clock rate 72000
Cisco_B(config-if)# bandwidth 72
Cisco_B(config-if)# ip address 200.100.100.9 255.255.255.252
Cisco_B(config-if)# no shutdown
Cisco_B(config-if)# exit
Cisco_B(config)# router ospf 100
Cisco_B(config-router)# network 200.100.100.0 0.0.0.255 area 0.0.0.0
```

This is exactly the same configuration that we had before the GRE Tunnel section.

6.2 Edge Router Configuration



Cisco X

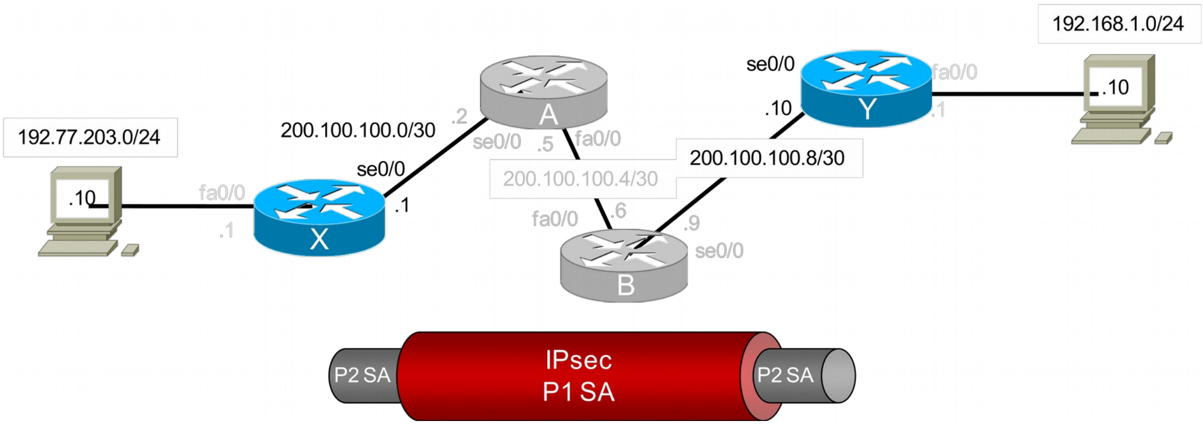
```
Router(config)# hostname Cisco_X
(config)# ip routing
Cisco_X(config)# int fa 0/0
Cisco_X(config-if)# ip address 192.77.203.1 255.255.255.0
Cisco_X(config-if)# no shutdown
Cisco_X(config-if)# exit
Cisco_X(config)# int se 0/0
Cisco_X(config-if)# clock rate 72000
Cisco_X(config-if)# bandwidth 72
Cisco_X(config-if)# ip address 200.100.100.1 255.255.255.252
Cisco_X(config-if)# no shutdown
Cisco_X(config-if)# exit
Cisco_X(config)# router ospf 100
Cisco_X(config-router)# network 200.100.100.0 0.0.0.255 area 0.0.0.0
Cisco_X(config-router)# network 192.77.203.0 0.0.0.255 area 0.0.0.0
Cisco_X(config-router)# passive-interface fa0/0
```

This is more or less the same configuration we had prior to the GRE Tunnel section except that OSPF will now also advertise the 192.77.203.0/24 network.

Cisco Y

```
Router(config)# hostname Cisco_Y
(config)# ip routing
Cisco_Y(config)# int fa 0/0
Cisco_Y(config-if)# ip address 192.77.203.1 255.255.255.0
Cisco_Y(config-if)# no shutdown
Cisco_Y(config-if)# exit
Cisco_Y(config)# int se 0/0
Cisco_Y(config-if)# clock rate 72000
Cisco_Y(config-if)# bandwidth 72
Cisco_Y(config-if)# ip address 200.100.100.1 255.255.255.252
Cisco_Y(config-if)# no shutdown
Cisco_Y(config-if)# exit
Cisco_Y(config)# router ospf 100
Cisco_Y(config-router)# network 200.100.100.0 0.0.0.255 area 0.0.0.0
Cisco_Y(config-router)# network 192.168.1.0 0.0.0.255 area 0.0.0.0
Cisco_Y(config-router)# passive-interface fa0/0
```

This is more or less the same configuration we had prior to the GRE Tunnel section except that OSPF will now also advertise the 192.168.1.0/24 network.



6.3 The ISAKMP Policy

Cisco X

```
Cisco_X(config)# crypto isakmp policy 1                 (1 – 10000 with 1 as the highest)
Cisco_X(config-isakmp)# encryption aes                  (AES / DES)
Cisco_X(config-isakmp)# hash sha                        (MD5 / SHA1)
Cisco_X(config-isakmp)# authentication pre-share
Cisco_X(config-isakmp)# lifetime 86400                  (86,400 seconds or one day)
Cisco_X(config-isakmp)# group 2                         (1 - 768-bit, 2 – 1024-bit, 5 – 1536-bit  Diffie-Hellman)
Cisco_X(config-isakmp)# exit
```

Cisco Y

```
Cisco_Y(config)# crypto isakmp policy 1                 (1 – 10000 with 1 as the highest)
Cisco_Y(config-isakmp)# encryption aes                  (AES / DES)
Cisco_Y(config-isakmp)# hash sha                        (MD5 / SHA1)
Cisco_Y(config-isakmp)# authentication pre-share
Cisco_Y(config-isakmp)# lifetime 86400                  (86,400 seconds or one day)
Cisco_Y(config-isakmp)# group 2                         (1 - 768-bit, 2 – 1024-bit, 5 – 1536-bit  Diffie-Hellman)
Cisco_Y(config-isakmp)# exit
```

Cisco X

```
Cisco_X(config)# crypto isakmp identity address
Cisco_X(config)# crypto isakmp key ipsec_secret address 200.100.100.10
```

Cisco Y

```
Cisco_Y(config)# crypto isakmp identity address
Cisco_Y(config)# crypto isakmp key ipsec_secret address 200.100.100.1
```

We must now define a transform set on each side. In this case we intend to use IPsec tunnel mode. In this mode IPsec encrypts the IP header and the payload, whereas transport mode only encrypts the IP payload. The IP addresses of the outer IP header are the tunnel endpoints, and the IP addresses of the encapsulated IP header are the ultimate source and destination addresses.

IPSec tunnel mode is useful for protecting traffic between different networks, when traffic must pass through an intermediate, untrusted network.

Cisco X

```
Cisco_X(config)# crypto ipsec transform-set ts1 ah-sha-hmac esp-aes
Cisco_X(cfg-ctypto-trans)# mode tunnel                          (transport | tunnel)
Cisco_X(cfg-ctypto-trans)# exit
```

Cisco Y

```
Cisco_Y(config)# crypto ipsec transform-set ts1 ah-sha-hmac esp-aes
Cisco_Y(cfg-ctypto-trans)# mode tunnel
Cisco_Y(cfg-ctypto-trans)# exit
```

We must now create a crypto map on each side.

Cisco X

```
Cisco_X(config)# access-list 110 permit ip 192.77.203.0 0.0.0.255 192.168.1.0 0.0.0.255

Cisco_X(config)# crypto map vpn 10 ipsec-isakmp          (ipsec-isakmp | ipsec-manual | dynamic)
Cisco_X(cfg-ctypto-map)# match address 110               (110 is the access-list number)
Cisco_X(cfg-ctypto-map)# set peer 200.100.100.10
Cisco_X(cfg-ctypto-map)# set transform-set ts1           (as defined earlier i.e. ts1)
Cisco_X(cfg-ctypto-map)# exit
```

Cisco Y

```
Cisco_Y(config)# access-list 110 permit ip 192.168.1.0 0.0.0.255 192.77.203.0 0.0.0.255

Cisco_Y(config)# crypto map vpn 10 ipsec-isakmp
Cisco_Y(cfg-ctypto-map)# match address 110
Cisco_Y(cfg-ctypto-map)# set peer 200.100.100.1
Cisco_Y(cfg-ctypto-map)# set transform-set ts1
Cisco_Y(cfg-ctypto-map)# exit
```

We must now assign the crypto map set to the physical public interface on each router.

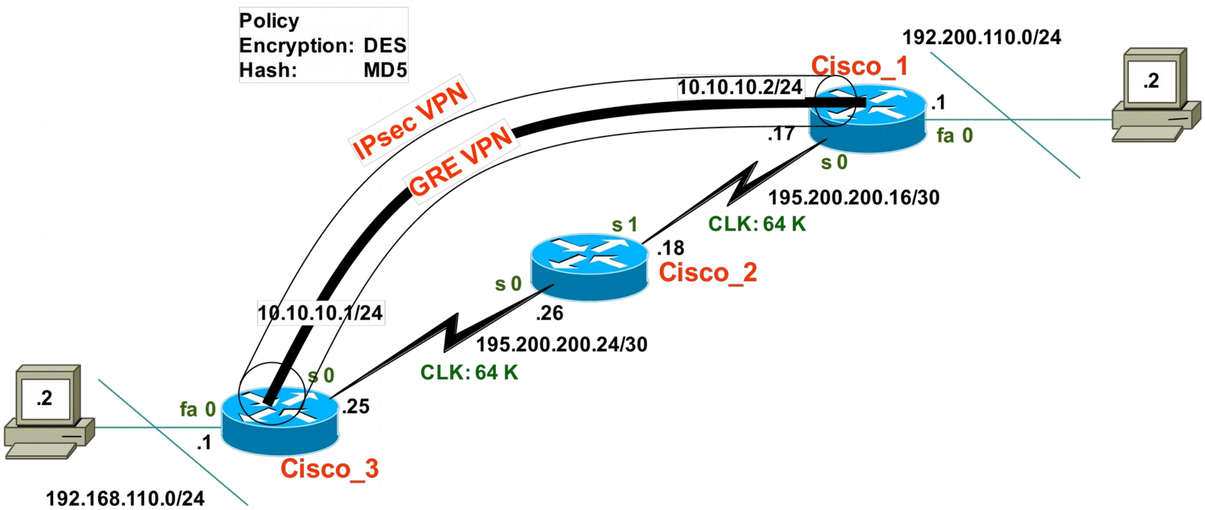
Cisco X

```
Cisco_X(config)# interface se 0/0
Cisco_X(config-if)# crypto map vpn
Cisco_X(config-if)# exit
```

Cisco Y

```
Cisco_Y(config)# interface se 0/0
Cisco_Y(config-if)# crypto map vpn
Cisco_Y(config-if)# exit
```

7 Lab Exercise - Configuring IPsec



7.1 Objective

- Practice building IPsec/GRE VPN Links.
- Practice testing IPsec/GRE VPN Links.

7.2 Background

Knowing how to configure routers for interconnecting LANs via IPsec/GRE VPN is an essential building block to you networking knowledge.

7.3 Lab Steps

Physically build network as shown.

7.4 Lab Commands

7.4.1 Cisco 2

```
Router(config)# hostname Cisco_2
Cisco_2(config)# ip routing
Cisco_2(config)# int se 0/0
Cisco_2(config-if)# clock rate 64000
Cisco_2(config-if)# bandwidth 64
Cisco_2(config-if)# ip address 195.200.200.26 255.255.255.252
Cisco_2(config-if)# no shutdown
Cisco_2(config-if)# exit
Cisco_2(config)# int se 0/1
Cisco_2(config-if)# clock rate 64000
Cisco_2(config-if)# bandwidth 64
Cisco_2(config-if)# ip address 195.20.200.18 255.255.255.252
Cisco_2(config-if)# no shutdown
Cisco_2(config-if)# exit
Cisco_2(config)# router ospf 100
Cisco_2(config-router)# network 195.200.200.0 0.0.0.255 area 0.0.0.0
Cisco_2(config-router)# exit
```

7.4.2 Cisco 1

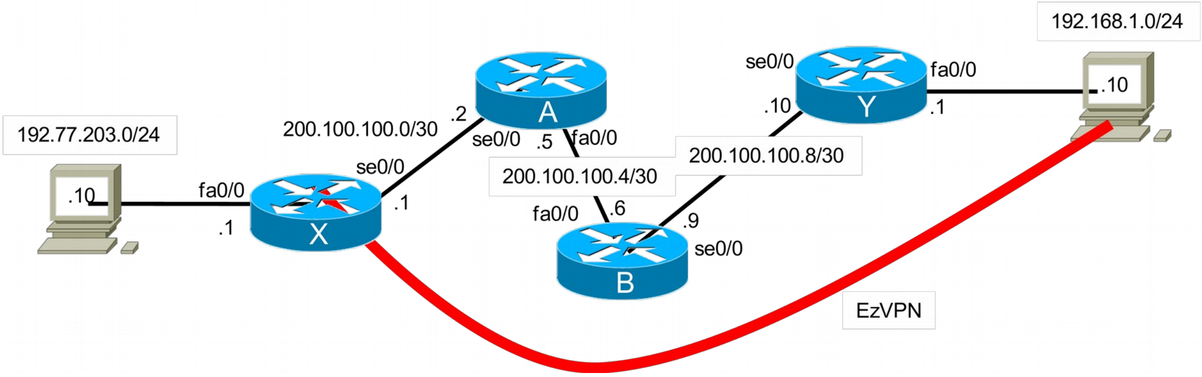
```
Router(config)# hostname Cisco_1
(config)# ip routing
Cisco_1(config)# int fa 0/0
Cisco_1(config-if)# ip address 192.200.110.1 255.255.255.0
Cisco_1(config-if)# no shutdown
Cisco_1(config-if)# exit
Cisco_1(config)# int se 0/0
Cisco_1(config-if)# clock rate 64000
Cisco_1(config-if)# bandwidth 64
Cisco_1(config-if)# ip address 195.200.200.17 255.255.255.252
Cisco_1(config-if)# no shutdown
Cisco_1(config-if)# exit
Cisco_1(config)# router ospf 100
Cisco_1(config-router)# network 195.200.200.0 0.0.0.255 area 0.0.0.0
Cisco_1(config-router)# passive-interface fa0/0
Cisco_1(config-router)# exit

Cisco_1(config)# int tunnel 0
Cisco_1(config-if)# ip address 10.10.10.2 255.255.255.0
Cisco_1(config-if)# tunnel source se0/0
Cisco_1(config-if)# tunnel destination 195.200.200.25
Cisco_1(config-if)# ip tcp adjust-mss 1436
Cisco_1(config-if)# tunnel key 1234
Cisco_1(config-if)# exit
Cisco_1(config)# ip route 192.168.110.0 255.255.255.0 10.10.10.1

Cisco_1(config)# crypto isakmp policy 1
Cisco_1(config-isakmp)# encryption des
Cisco_1(config-isakmp)# hash md5
Cisco_1(config-isakmp)# authentication pre-share
Cisco_1(config-isakmp)# lifetime 86400
Cisco_1(config-isakmp)# group 2
Cisco_1(config-isakmp)# exit
Cisco_1(config)# crypto isakmp identity address
Cisco_1(config)# crypto isakmp key ipsec_secret address 195.200.200.25

Cisco_1(config)# crypto ipsec transform-set ts1 ah-sha-hmac esp-aes
Cisco_1(cfg-ctypto-trans)# mode transport
Cisco_1(cfg-ctypto-trans)# exit

Cisco_1(config)# access-list 110 permit gre host 195.200.200.17 host 195.200.200.25

Cisco_1(config)# crypto map vpn 10 ipsec-isakmp
Cisco_1(cfg-ctypto-map)# match address 110
Cisco_1(cfg-ctypto-map)# set peer 195.200.200.25
Cisco_1(cfg-ctypto-map)# set transform-set ts1
Cisco_1(cfg-ctypto-map)# exit

Cisco_1(config)# interface Tunnel0
Cisco_1(config-if)# crypto map vpn
Cisco_1(config-if)# exit
Cisco_1(config)# interface se 0/0
Cisco_1(config-if)# crypto map vpn
Cisco_1(config-if)# exit
```

7.4.3 Cisco 3

```
Router(config)# hostname Cisco_3
(config)# ip routing
Cisco_3(config)# int fa 0/0
Cisco_3(config-if)# ip address 192.168.110.1 255.255.255.0
Cisco_3(config-if)# no shutdown
Cisco_3(config-if)# exit
Cisco_3(config)# int se 0/0
Cisco_3(config-if)# clock rate 64000
Cisco_3(config-if)# bandwidth 64
Cisco_3(config-if)# ip address 195.200.200.25 255.255.255.252
Cisco_3(config-if)# no shutdown
Cisco_3(config-if)# exit
Cisco_3(config)# router ospf 100
Cisco_3(config-router)# network 195.200.200.0 0.0.0.255 area 0.0.0.0
Cisco_3(config-router)# passive-interface fa0/0
Cisco_3(config-router)# exit

Cisco_3(config)# int tunnel 0
Cisco_3(config-if)# ip address 10.10.10.1 255.255.255.0
Cisco_3(config-if)# tunnel source se0/0
Cisco_3(config-if)# tunnel destination 195.200.200.17
Cisco_3(config-if)# ip tcp adjust-mss 1436
Cisco_3(config-if)# tunnel key 1234
Cisco_3(config-if)# exit
Cisco_3(config)# ip route 192.200.110.0 255.255.255.0 10.10.10.2

Cisco_3(config)# crypto isakmp policy 1
Cisco_3(config-isakmp)# encryption des
Cisco_3(config-isakmp)# hash md5
Cisco_3(config-isakmp)# authentication pre-share
Cisco_3(config-isakmp)# lifetime 86400
Cisco_3(config-isakmp)# group 2
Cisco_3(config-isakmp)# exit
Cisco_3(config)# crypto isakmp identity address
Cisco_3(config)# crypto isakmp key ipsec_secret address 195.200.200.17

Cisco_3(config)# crypto ipsec transform-set ts1 ah-sha-hmac esp-aes
Cisco_3(cfg-ctypto-trans)# mode transport
Cisco_3(cfg-ctypto-trans)# exit

Cisco_3(config)# access-list 110 permit gre host 195.200.200.25 host 195.200.200.17

Cisco_3(config)# crypto map vpn 10 ipsec-isakmp
Cisco_3(cfg-ctypto-map)# match address 110
Cisco_3(cfg-ctypto-map)# set peer 195.200.200.17
Cisco_3(cfg-ctypto-map)# set transform-set ts1
Cisco_3(cfg-ctypto-map)# exit

Cisco_3(config)# interface Tunnel0
Cisco_3(config-if)# crypto map vpn
Cisco_3(config-if)# exit
Cisco_3(config)# interface se 0/0
Cisco_3(config-if)# crypto map vpn
Cisco_3(config-if)# exit
```

8 EzVPN

Cisco Easy VPN makes use of the Cisco Unity Client protocol. This is designed to allow remote PC users with the Cisco VPN Client access their network securely via a Cisco Router, Firewall or Concentrator.



The EzVPN client attempts to establish an IPsec connection to the EzVPN Server, once this is established the Xauth feature allows the EzVPN client to wait for a “username/password” challenge after the IKE SA has been established.

The end user responds to the challenge, the response is forwarded to the IPsec peers for an additional level of authentication by the router, the router can authenticate this based on the AAA configuration, local user or RADIUS/TACACS Server.

8.1 Split Tunnelling

Unless split tunnelling is enabled all user traffic is through the tunnel. However by assigning an ACL to determine the traffic to be tunnelled in the client configuration group only traffic matching the ACL will be tunnelled and all other traffic will bypass the tunnel directly to the network unencrypted. The ACL is dynamically loaded on the Easy VPN Client, and specifies exactly the networks to be permitted for encryption. Split tunnelling uses the hub router resources efficiently, freeing the server bandwidth for additional VPN clients.

8.2 Building the testbed - Routing

Use the scripts below to re-establish the testbed to a point where it is only routing and without any VPNs configured.

Cisco A

```
hostname Cisco_A
interface FastEthernet0/0
ip address 200.100.100.5 255.255.255.252
no shutdown
exit
interface Serial0/0/0
bandwidth 128
clock rate 128000
ip address 200.100.100.2 255.255.255.252
no shutdown
exit
router ospf 100
network 200.100.100.0 0.0.0.255 area 0.0.0.0
exit
```

Cisco B

```
hostname Cisco_B
interface FastEthernet0
ip address 200.100.100.6 255.255.255.252
no shutdown
exit
interface Serial0
bandwidth 128
clock rate 128000
ip address 200.100.100.9 255.255.255.252
no shutdown
exit
router ospf 100
network 200.100.100.0 0.0.0.255 area 0.0.0.0
exit
```

Cisco X

```
hostname Cisco_X
interface FastEthernet0/0
ip address 192.77.203.1 255.255.255.0
no shutdown
exit
interface Serial0/2/0
bandwidth 128
clock rate 128000
ip address 200.100.100.1 255.255.255.252
no shutdown
exit
router ospf 100
network 200.100.100.0 0.0.0.255 area 0.0.0.0
network 192.77.203.0 0.0.0.255 area 0.0.0.0
passive-interface FastEthernet0/0
exit
```

Cisco Y

```
hostname Cisco_Y
interface FastEthernet0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
interface Serial0/0/0
bandwidth 128
clock rate 128000
ip address 200.100.100.10 255.255.255.252
no shutdown
exit
router ospf 100
network 200.100.100.0 0.0.0.255 area 0.0.0.0
network 192.168.1.0 0.0.0.255 area 0.0.0.0
passive-interface FastEthernet0/0
exit
```

8.2.1 Test

Test the system by pinging from the PC 192.168.1.10 to 192.77.203.10.

If successful save the configurations and remove the 192.77.203.0/24 network from the VPN Server router. This network does not need to be advertised as it is the private secure network. The VPN is established to the other interface on the VPN Server. When the client connects it will be assigned an IP address for the VPN Client Adapter from the pool of addresses assigned for remote users.

Cisco X

```
router ospf 100
no network 192.77.203.0 0.0.0.255 area 0.0.0.0
exit
```

8.3 Configure the EzVPN Server

The AAA new model is enabled and we specify that login USERS are authenticated locally (not on a RADIUS or TACACS Server). We also define that the authorisation CLIENT-GROUP is also defined locally. The username and password to authenticate the remote user is generated here as well. I also define the IP Doman name.

```
Cisco_X(config)# aaa new-model

Cisco_X(config)# aaa authentication login USERS local
Cisco_X(config)# aaa authorization network CLIENT-GROUP local
Cisco_X(config)# username ezuser pass 0 ezpass
Cisco_X(config)# ip domain-name c2s.ie
```

The ISAKMP/IKE Phase I policy for the VPN Client is defined. Also configured here is the keepalive or Dead Peer Detection (DPD). In this case the router will send DPD messages to the peer every 20 seconds and should a DPD fail it will send another after 3 seconds. Depending on the version of IOS used it maybe useful to enable fragmentation of large IKE packets to avoid fragmentation at the UDP layer.

```
; Cisco_X(config)#crypto isakmp fragmentation    (Not applicable on all IOS versions)
Cisco_X(config)# crypto isakmp keepalive 20 3

Cisco_X(config)# crypto isakmp policy 10
Cisco_X(config-isakmp)# encr aes 128
Cisco_X(config-isakmp)# hash sha
Cisco_X(config-isakmp)# authentication pre-share
Cisco_X(config-isakmp)# group 2
Cisco_X(config-isakmp)# exit
```

Create IP Address Pool on SECURE-NET, these addresses will be assigned to users connecting from VPN Clients.

```
Cisco_X(config)# ip local pool DYNAMIC-POOL 192.77.203.100 192.77.203.150
```

If Split tunnelling is to be used an Access List (ACL) is needed and will be assigned to the client configuration group.

```
Cisco_X(config)# ip access-list extended VPN-ACL
Cisco_X(config-ext-nacl)# permit ip 192.77.203.0 0.0.0.255 any
Cisco_X(config-ext-nacl)# exit
```

ISAKMP/IKE Client Configuration. The group CLIENT-GROUP defines the parameters for the VPN Client connection and the key is the group key used when creating the profile on the VPN Client. If split tunnelling is being used the ACL is included here to identify the traffic to be tunnelled.

```
Cisco_X(config)# crypto isakmp client configuration address-pool local DYNAMIC-POOL
Cisco_X(config)# crypto isakmp xauth timeout 60

Cisco_X(config)# crypto isakmp client configuration group CLIENT-GROUP
Cisco_X(config-isakmp-group)# key 0 cgpass
Cisco_X(config-isakmp-group)# domain c2s.ie
Cisco_X(config-isakmp-group)# pool DYNAMIC-POOL
Cisco_X(config-isakmp-group)# acl VPN-ACL
Cisco_X(config-isakmp-group)# exit
```

A transform set is defined for the Remote data connections.

```
Cisco_X(config)# crypto ipsec transform-set TRANSFORM1 esp-aes esp-sha-hmac
Cisco_X(cfg-crypto-trans)# exit
```

A dynamic cryptographic map is created for the remote access users and the Transform Set is mapped to it. The reverse-route command is used to create source proxy information for the crypto map.

```
Cisco_X(config)# crypto dynamic-map DYNAMIC-MAP 10
Cisco_X(config-crypto-map)# set transform-set TRANSFORM1
Cisco_X(config-crypto-map)# reverse-route
Cisco_X(config-crypto-map)# exit
```

Static cryptographic map is created XAUTH enabled and references the AAA authentication login command to determine where to find the user accounts.

```
Cisco_X(config)# crypto map STATIC-MAP client authentication list USERS
```

The remote access group is associated with the cryptographic map. This tells the router where to find the group attributes. This command references the AAA authorization network statements that tell the router that the group is defined locally.

```
Cisco_X(config)# crypto map STATIC-MAP isakmp authorization  list CLIENT-GROUP
```

If the Remote clients can initiate IKE Mode Config (Cisco clients 4.x and above), then the VPN Server is configured to respond to IKE Mode Config queries.

```
Cisco_X(config)# crypto map STATIC-MAP client configuration address respond
```

The dynamic cryptographic map is referenced in a static cryptographic map.

```
Cisco_X(config)# crypto map STATIC-MAP 100 ipsec-isakmp dynamic DYNAMIC-MAP
```

Static Cryptographic map is applied to the public interface connected to the NON-TRUSTED-NET. A description can be added to prevent confusion.

```
Cisco_X(config)# int se0/0
Cisco_X(config-if)# description NON-TRUSTED-NET
Cisco_X(config-if)# crypto map STATIC-MAP
Cisco_X(config-if)# exit
```

A description can be added to the SECURE-NET also as an identifier.

```
Cisco_X(config)# int fastethernet 0/0
Cisco_X(config-if)# description SECURE-NET
Cisco_X(config-if)# exit
```

8.3.1 EzVPN Command Summary Script

```
aaa new-model
aaa authentication login USERS local
aaa authorization network CLIENT-GROUP local
username ezuser pass 0 ezpass
ip domain-name c2s.ie
crypto isakmp fragmentation
crypto isakmp keepalive 20 3
crypto isakmp policy 10
encr aes 128
hash sha
authentication pre-share
group 2
exit
ip local pool DYNAMIC-POOL 192.77.203.100 192.77.203.150
ip access-list extended VPN-ACL
permit ip 192.77.203.0 0.0.0.255 any
exit
crypto isakmp client configuration address-pool local DYNAMIC-POOL
crypto isakmp xauth timeout 60
crypto isakmp client configuration group CLIENT-GROUP
key 0 cgpass
domain c2s.ie
pool DYNAMIC-POOL
acl VPN-ACL
exit
crypto ipsec transform-set TRANSFORM1 esp-aes esp-sha-hmac
exit
crypto dynamic-map DYNAMIC-MAP 10
set transform-set TRANSFORM1
reverse-route
exit
crypto map STATIC-MAP client authentication list USERS
crypto map STATIC-MAP isakmp authorization list CLIENT-GROUP
crypto map STATIC-MAP client configuration address respond
crypto map STATIC-MAP 100 ipsec-isakmp dynamic DYNAMIC-MAP
int se0/0
description NON-TRUSTED-NET
crypto map STATIC-MAP
exit
int fastethernet 0/0
description SECURE-NET
exit
```

8.4 Configure the Cisco VPN Client

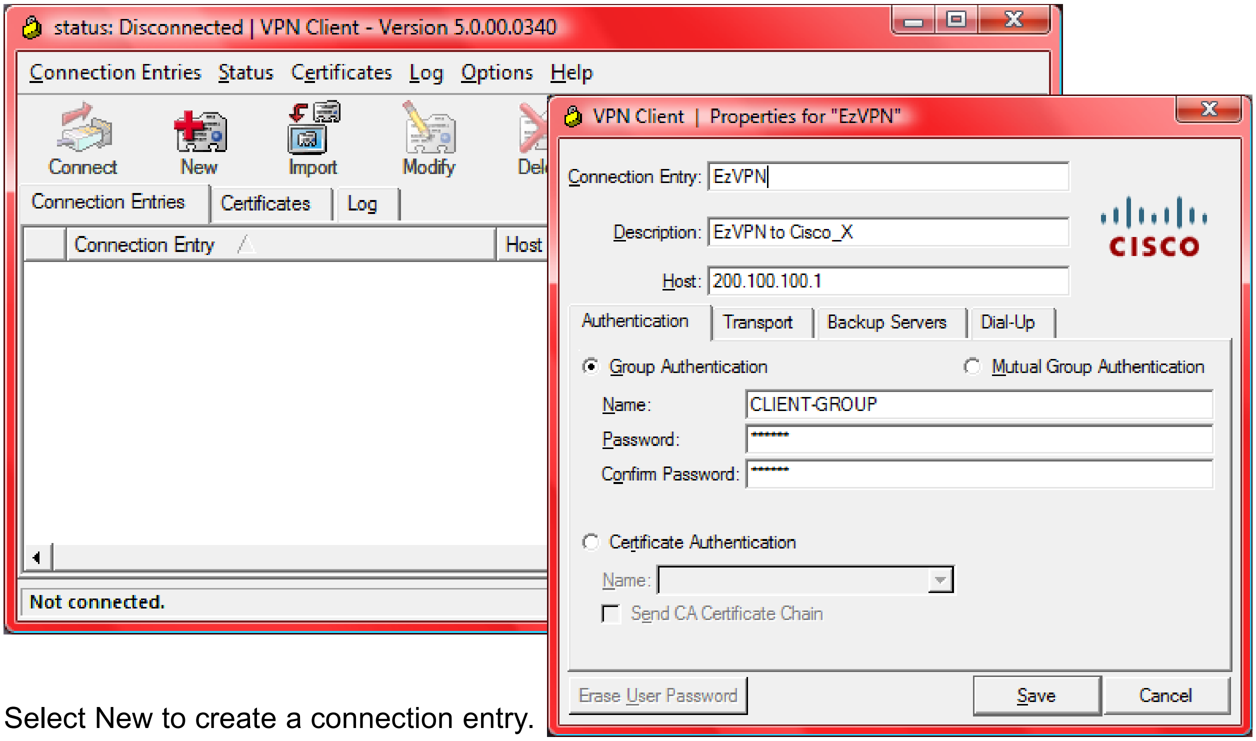
The Cisco VPN Client comes in versions for Linux, Sun Solaris and Microsoft Windows. Install the application and reboot the computer.

On Sun and Linux execute the application with the command vpnclient:

```
jdoe@ubuntu:/~$ vpnclient
```

To start the client on Microsoft Windows select:

Start → Progams → Cisco Systems VPN Client → VPN Client



status: Disconnected | VPN Client - Version 5.0.00.0340

Not connected.

VPN Client | Properties for "EzVPN"

Connection Entry: EzVPN

Description: EzVPN to Cisco_X

Host: 200.100.100.1

● Group Authentication ○ Mutual Group Authentication

Name: CLIENT-GROUP

Password: *******

Confirm Password: *******

○ Certificate Authentication

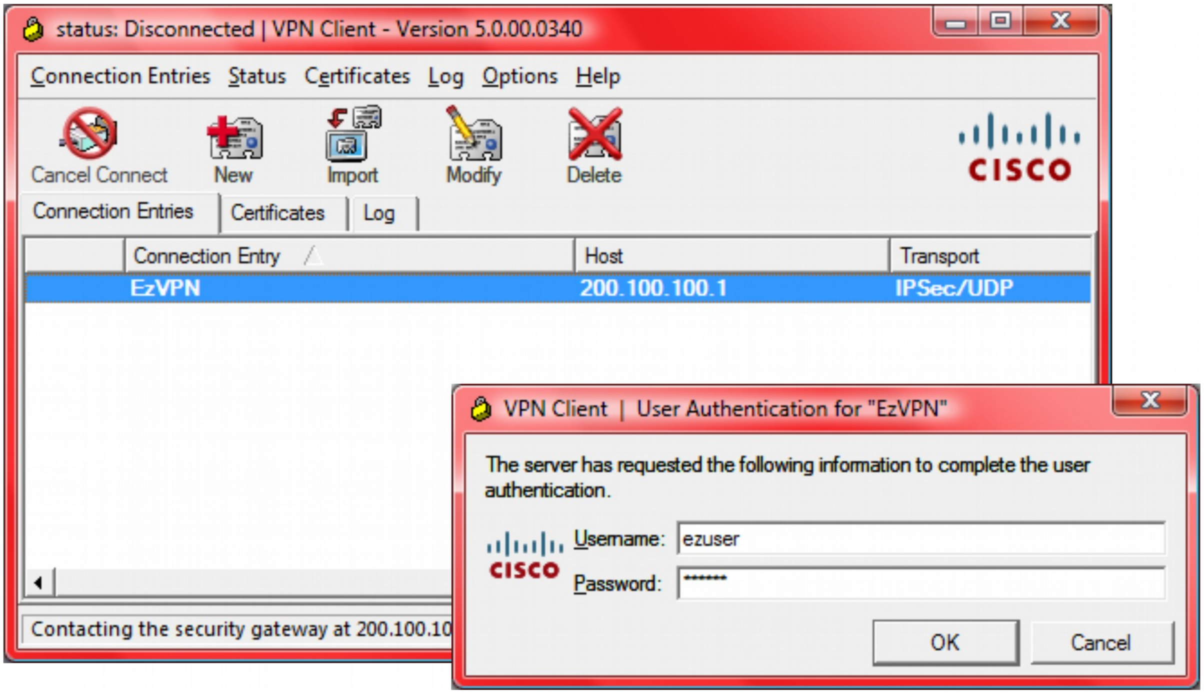
Name:

□ Send CA Certificate Chain

Select New to create a connection entry.

- Place a connection name in the Connection Entry dialogue.
- Write a helpful Description.
- Add in the Host: the IP address of the NON-TRUSTED-NET interface on Cisco_X.
- In the Authentication Tab enter the client configuration group name ***CLIENT-GROUP*** and in the Password fields enter the associated password: ***cgpass***
- ***Save*** the configuration.

8.5 Connecting to the VPN Server



status: Disconnected | VPN Client - Version 5.0.00.0340

	Connection Entry	Host	Transport
	EzVPN	200.100.100.1	IPSec/UDP

Contacting the security gateway at 200.100.10

VPN Client | User Authentication for "EzVPN"

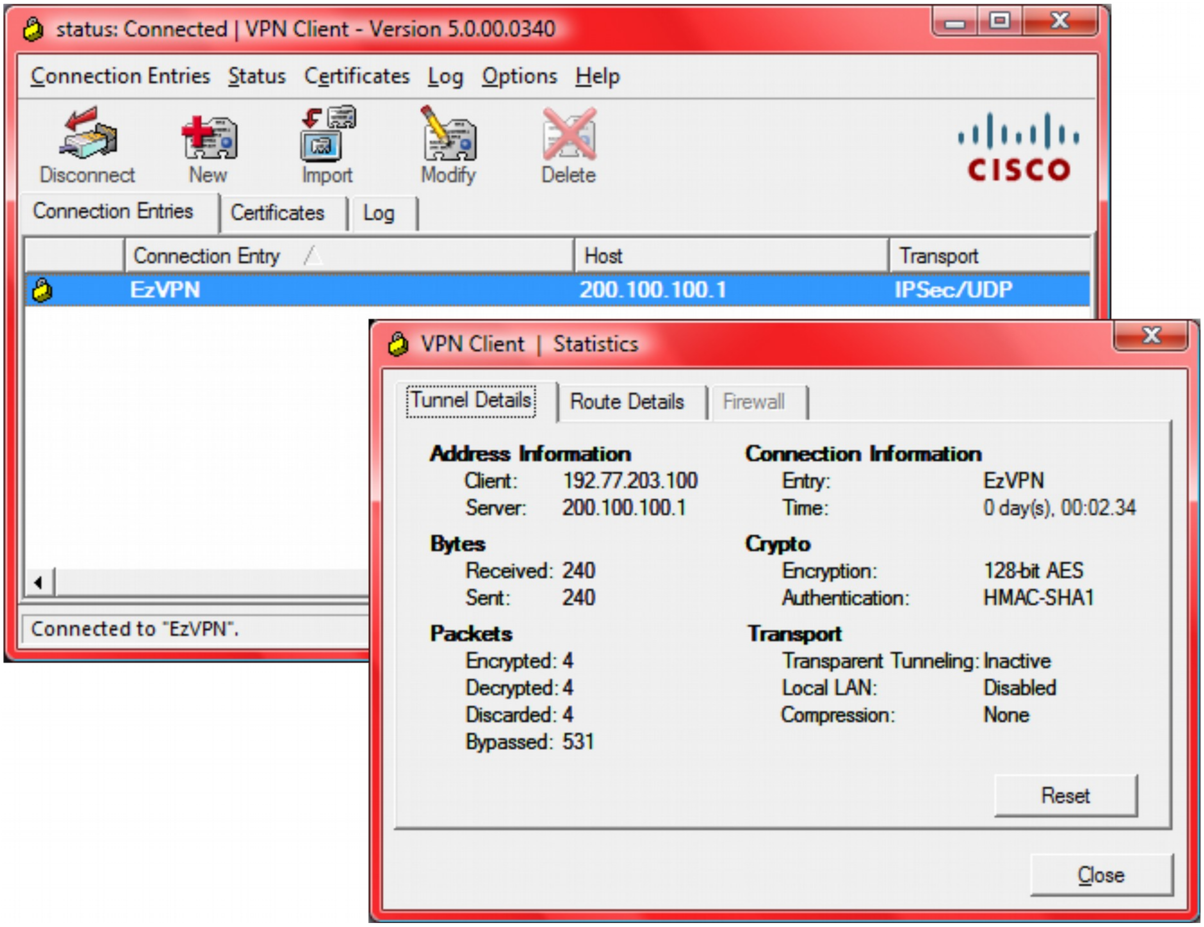
The server has requested the following information to complete the user authentication.

Username: ezuser

Password: ******

Double click on your new connection and you will be presented with a Username/Password dialogue courtesy of Xauth. Enter the username and password configured in this case locally in the router though it would be more typical to have it on a AAA Server like RADIUS or TACACS. Clock ***OK***.

8.6 Confirming the connection



status: Connected | VPN Client - Version 5.0.00.0340

Connection Entry	Host	Transport
EzVPN	200.100.100.1	IPSec/UDP

Connected to "EzVPN".

VPN Client | Statistics

Address Information
Client: 192.77.203.100
Server: 200.100.100.1

Bytes
Received: 240
Sent: 240

Packets
Encrypted: 4
Decrypted: 4
Discarded: 4
Bypassed: 531

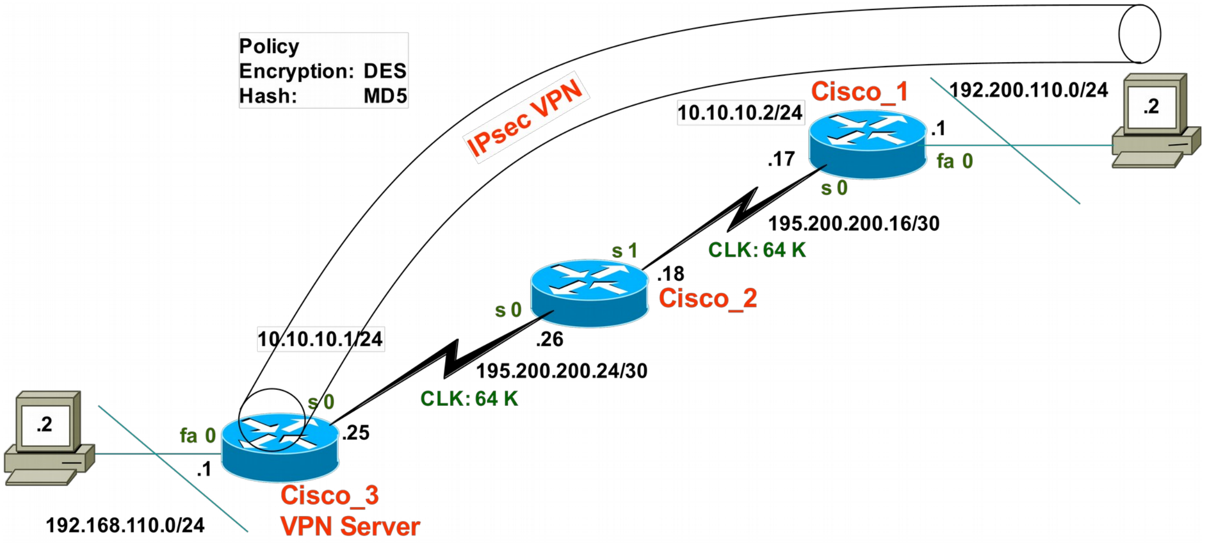
Connection Information
Entry: EzVPN
Time: 0 day(s), 00:02.34

Crypto
Encryption: 128-bit AES
Authentication: HMAC-SHA1

Transport
Transparent Tunneling: Inactive
Local LAN: Disabled
Compression: None

After a little while the Cisco VPN Client will show connected to your connection. Selecting ***Status → Statistics*** will give detail on the connection.

9 Lab Exercise - Configuring IPsec VPN Server/Client



9.1 Objective

- Practice building IPsec VPN Server/Client Links.
- Practice testing IPsec VPN Server/Client Links.

9.2 Background

Knowing how to configure routers for connecting remote VPN Clients via IPsec VPN Server/Client is an essential building block to you networking knowledge.

9.3 Lab Steps

Physically build network as shown.

9.4 Lab Commands

9.4.1 Cisco 2

```
Router(config)# hostname Cisco_2
Cisco_2(config)# ip routing
Cisco_2(config)# int se 0/0
Cisco_2(config-if)# clock rate 64000
Cisco_2(config-if)# bandwidth 64
Cisco_2(config-if)# ip address 195.200.200.26 255.255.255.252
Cisco_2(config-if)# no shutdown
Cisco_2(config-if)# exit
Cisco_2(config)# int se 0/1
Cisco_2(config-if)# clock rate 64000
Cisco_2(config-if)# bandwidth 64
Cisco_2(config-if)# ip address 195.20.200.18 255.255.255.252
Cisco_2(config-if)# no shutdown
Cisco_2(config-if)# exit
Cisco_2(config)# router ospf 100
Cisco_2(config-router)# network 195.200.200.0 0.0.0.255 area 0.0.0.0
Cisco_2(config-router)# exit
```

9.4.2 Cisco 3

```
Router(config)# hostname Cisco_3
(config)# ip routing
Cisco_3(config)# int fa 0/0
Cisco_3(config-if)# ip address 192.168.110.1 255.255.255.0
Cisco_3(config-if)# no shutdown
Cisco_3(config-if)# exit
Cisco_3(config)# int se 0/0
Cisco_3(config-if)# clock rate 64000
Cisco_3(config-if)# bandwidth 64
Cisco_3(config-if)# ip address 195.200.200.25 255.255.255.252
Cisco_3(config-if)# no shutdown
Cisco_3(config-if)# exit
Cisco_3(config)# router ospf 100
Cisco_3(config-router)# network 195.200.200.0 0.0.0.255 area 0.0.0.0
Cisco_3(config-router)# passive-interface fa0/0
Cisco_3(config-router)# exit
```

9.4.3 Cisco 1

```
Router(config)# hostname Cisco_1
(config)# ip routing
Cisco_1(config)# int fa 0/0
Cisco_1(config-if)# ip address 192.200.110.1 255.255.255.0
Cisco_1(config-if)# no shutdown
Cisco_1(config-if)# exit
Cisco_1(config)# int se 0/0
Cisco_1(config-if)# clock rate 64000
Cisco_1(config-if)# bandwidth 64
Cisco_1(config-if)# ip address 195.200.200.17 255.255.255.252
Cisco_1(config-if)# no shutdown
Cisco_1(config-if)# exit
Cisco_1(config)# router ospf 100
Cisco_1(config-router)# network 195.200.200.0 0.0.0.255 area 0.0.0.0
Cisco_1(config-router)# network 192.200.110.0 0.0.0.255 area 0.0.0.0
Cisco_1(config-router)# passive-interface fa0/0
Cisco_1(config-router)# exit

Cisco_1(config)# aaa new-model
Cisco_1(config)# aaa authentication login USERS local
Cisco_1(config)# aaa authorization network CLIENT-GROUP local
Cisco_1(config)# aaa session-id common
Cisco_1(config)# username ezuser pass 0 ezpass
Cisco_1(config)# ip domain-name c2s.ie

Cisco_1(config)#crypto isakmp fragmentation
Cisco_1(config)# crypto isakmp keepalive 20 3
Cisco_1(config)# crypto isakmp policy 10
Cisco_1(config-isakmp)# encr 3des
Cisco_1(config-isakmp)# hash md5
Cisco_1(config-isakmp)# authentication pre-share
Cisco_1(config-isakmp)# group 2
Cisco_1(config-isakmp)# exit

Cisco_1(config)# ip local pool DYNAMIC-POOL 192.168.110.100 192.168.110.150
Cisco_1(config)# ip access-list extended VPN-ACL
Cisco_1(config-ext-nacl)# permit ip 192.168.110.0 0.0.0.255 any
Cisco_1(config-ext-nacl)# exit

Cisco_1(config)# crypto isakmp client configuration address-pool local DYNAMIC-POOL
Cisco_1(config)# crypto isakmp xauth timeout 60
Cisco_1(config)# crypto isakmp client configuration group CLIENT-GROUP
Cisco_1(config-isakmp-group)# key 0 cgpass
Cisco_1(config-isakmp-group)# domain c2s.ie
Cisco_1(config-isakmp-group)# pool DYNAMIC-POOL
Cisco_1(config-isakmp-group)# acl VPN-ACL
Cisco_1(config-isakmp-group)# exit

Cisco_1(config)# crypto ipsec transform-set TRANSFORM1 esp-3des esp-md5-hmac
Cisco_1(cfg-crypto-trans)# exit
```

```
Cisco_1(config)# crypto dynamic-map DYNAMIC-MAP 10
Cisco_1(config-crypto-map)# set transform-set TRANSFORM1
Cisco_1(config-crypto-map)# reverse-route
Cisco_1(config-crypto-map)# exit

Cisco_1(config)# crypto map STATIC-MAP client authentication list USERS
Cisco_1(config)# crypto map STATIC-MAP isakmp authorization list CLIENT-GROUP
Cisco_1(config)# crypto map STATIC-MAP client configuration address respond
Cisco_1(config)# crypto map STATIC-MAP 100 ipsec-isakmp dynamic DYNAMIC-MAP

Cisco_1(config)# int se0/0
Cisco_1(config-if)# description NON-TRUSTED-NET
Cisco_1(config-if)# crypto map STATIC-MAP
Cisco_1(config-if)# exit
Cisco_1(config)# int fastethernet 0/0
Cisco_1(config-if)# description SECURE-NET
Cisco_1(config-if)# exit
```

10 Troubleshooting IPsec

10.1 Cryptographic Sessions

General view of Cryptographic sessions by interface. Each cryptographic session is a set of IPSec flows between two cryptographic endpoints. If the two cryptographic endpoints use IKE as the keying protocol, they are IKE peers to each other. A cryptographic session consists of one IKE security association for control traffic and at least two IPSec security associations for data traffic, one per each direction. During the rekeying phases there may be duplicated IKE security associations (SAs) and IPSec SAs. This can also happen if there are simultaneous setup requests from each side.

```
Cisco_X# show crypto session
Crypto session current status

Interface: Serial0/2/0
Session status: UP-ACTIVE
Peer: 200.100.100.10 port 500
  IKE SA: local 200.100.100.1/500 remote 200.100.100.10/500 Active
  IPSEC FLOW: permit ip 192.77.203.0/255.255.255.0 192.168.1.0/255.255.255.0
        Active SAs: 4, origin: crypto map
```

Here we can see:

- Router Interface
- IKE peer description
- IKE SAs that are associated with the peer
    - These IKE SAs create the IPSec SAs
- IPSec SAs serving the flows of a session

```
Cisco_X# show crypto session detail
Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, X - IKE Extended Authentication
F - IKE Fragmentation

Interface: Serial0/2/0
Uptime: 00:59:52
Session status: UP-ACTIVE
Peer: 200.100.100.10 port 500 fvrf: (none) ivrf: (none)
      Phase1_id: 200.100.100.10
      Desc: (none)
  IKE SA: local 200.100.100.1/500 remote 200.100.100.10/500 Active
          Capabilities:(none) connid:1002 lifetime:22:59:02
  IPSEC FLOW: permit ip 192.77.203.0/255.255.255.0 192.168.1.0/255.255.255.0
        Active SAs: 4, origin: crypto map
        Inbound:  #pkts dec'ed 2000 drop 0 life (KB/Sec) 4423141/3409
        Outbound: #pkts enc'ed 2079 drop 15 life (KB/Sec) 4423141/3409
```

10.2 Check Secure Associations (SA) State

A good starter to is to check the Secure Associations (SA).

Cisco_X# **sh crypto ipsec sa**

```
interface: Serial0/2/0
    Crypto map tag: vpn, local addr 200.100.100.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (192.77.203.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   current_peer 200.100.100.10 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 1657, #pkts encrypt: 1657, #pkts digest: 1657
    #pkts decaps: 1578, #pkts decrypt: 1578, #pkts verify: 1578
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 15, #recv errors 0

     local crypto endpt.: 200.100.100.1, remote crypto endpt.: 200.100.100.10
     path mtu 1500, ip mtu 1500, ip mtu idb Serial0/2/0
     current outbound spi: 0x5DC99064(1573490788)

     inbound esp sas:
      spi: 0x8C70A116(2356191510)
        transform: esp-aes ,
        in use settings ={Tunnel, }
        conn id: 2003, flow_id: FPGA:3, crypto map: vpn
        sa timing: remaining key lifetime (k/sec): (4502321/2355)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     inbound ah sas:
      spi: 0xC9BA0490(3384411280)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2003, flow_id: FPGA:3, crypto map: vpn
        sa timing: remaining key lifetime (k/sec): (4502321/2355)
        replay detection support: Y
        Status: ACTIVE

     inbound pcp sas:

     outbound esp sas:
      spi: 0x5DC99064(1573490788)
        transform: esp-aes ,
        in use settings ={Tunnel, }
        conn id: 2004, flow_id: FPGA:4, crypto map: vpn
        sa timing: remaining key lifetime (k/sec): (4502320/2355)
        IV size: 16 bytes
        replay detection support: Y
        Status: ACTIVE

     outbound ah sas:
      spi: 0xD974B0D6(3648303318)
        transform: ah-sha-hmac ,
        in use settings ={Tunnel, }
        conn id: 2004, flow_id: FPGA:4, crypto map: vpn
        sa timing: remaining key lifetime (k/sec): (4502320/2355)
        replay detection support: Y
        Status: ACTIVE

     outbound pcp sas:
```

10.3 Check ISAKMP SA State

check the Internet Key Exchange (IKE) security associations (SAs) at a peer. Typically the Internet Security Association and Key Management Protocol (ISAKMP) SA will be in its quiescent state (QM_IDLE). The SA can also show in Main Mode state where it has NO_STATE identifying that the ISAKMP SA has been created, but nothing else has happened yet. It could also be in SETUP indicating that the peers have agreed on parameters for the ISAKMP SA or it could be in KEY EXCHange where the peers have exchanged Diffie-Hellman public keys and have generated a shared secret. Finally the KEY AUTHentication main mode state where the ISAKMP SA has been authenticated. A similiar set of states exist for Aggressive and Quick modes.

```
Cisco_X# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id slot status
200.100.100.1   200.100.100.10  QM_IDLE           1002    0 ACTIVE

IPv6 Crypto ISAKMP SA

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

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

1002  200.100.100.1   200.100.100.10           ACTIVE aes  sha  psk  1  23:40:0
       Engine-id:Conn-id =  SW:2

IPv6 Crypto ISAKMP SA
```

10.4 Clearing Crypto Sessions

The ***clear crypto session*** command gives the ability to delete all IKE and IPsec SAs associated with all peers or a specific peer.

```
Cisco_X# clear crypto session ?
            active     Clears HA-enabled crypto sessions in the active state
            fvrf       Front-door mapping to MPLS Virtual Routing and Forwarding (VRF)
            isakmp     Clear crypto sessions belonging to the group
            ivrf       Inside MPLS VRF
            local      Clear crypto sessions for a local crypto endpoint
            remote     Clear crypto sessions for a remote IKE peer
            standby    Clears HA-enabled crypto sessions in the standby state
            username   Clear crypto sessions of a user

Cisco_X# clear crypto session active

Cisco_X# show crypto session
Crypto session current status

Interface: Serial0/2/0
Session status: UP-ACTIVE
Peer: 200.100.100.10 port 500
  IKE SA: local 200.100.100.1/500 remote 200.100.100.10/500 Active
  IPSEC FLOW: permit ip 192.77.203.0/255.255.255.0 192.168.1.0/255.255.255.0
        Active SAs: 4, origin: crypto map
```

When executed this command tears down the session(s) which will then have a "DOWN-NEGOTIATING" status in the ***show crypto session*** command output. This indicates that the SAs are either completely down or in the process of being brought back up.

10.5 Invalid Secure Parameter Index (SPI)

Sometimes after rebooting the routers it is found that the Secure Associations are marked as IDLE and a message like this appears on the console of one side. This occurs when one IPSec peer disappears like when a reboot occurs for some reason. Because the receiving peer is completely reset, it loses its IKE SA with the other peer. When an IPSec peer receives a packet for which it cannot find an SA, it sends an IKE "*INVALID SPI NOTIFY*" message to the originator. This notification is sent using the IKE SA. If there is no IKE SA available, the receiving peer drops the packet.

```
%CRYPTO-4-RECVD_PKT_INV_SPI: decaps: rec'd IPSEC packet has invalid spi for destaddr=200.100.100.10, prot=51,  spi= 0x8591
```

To recover execute the command:

```
Cisco_X(config)# crypto isakmp invalid-spi-recovery
```

10.6 Debug commands

Displays errors during Phase 1 IKE ISAKMP.

```
Cisco_X(config)# debug crypto isakmp
```

Displays errors during Phase 2.

```
Cisco_X(config)# debug crypto ipsec
```

Displays information from the cryptographic engine

```
Cisco_X(config)# debug crypto engine
```

Displays information from the crypto engine.

This page is intentionally blank