Virtual Private Networks (VPN's)

VPN overview

A VPN carries private traffic over a public network using advanced encryption and tunnels to protect:

- Confidentiality of information
- Integrity of data
- Authentication of users

- Virtual Private Network (VPN) is defined as network connectivity deployed on a shared infrastructure with the same policies and security as a private network.
- A VPN can be between two end systems, or it can be between two or more networks.
- A VPN can be built using tunnels and encryption. VPNs can occur at any layer of the OSI protocol stack.
- A VPN is an alternative WAN infrastructure that replaces or augments existing private networks that use leased-line or enterprise-owned Frame Relay or ATM networks.

VPN overview

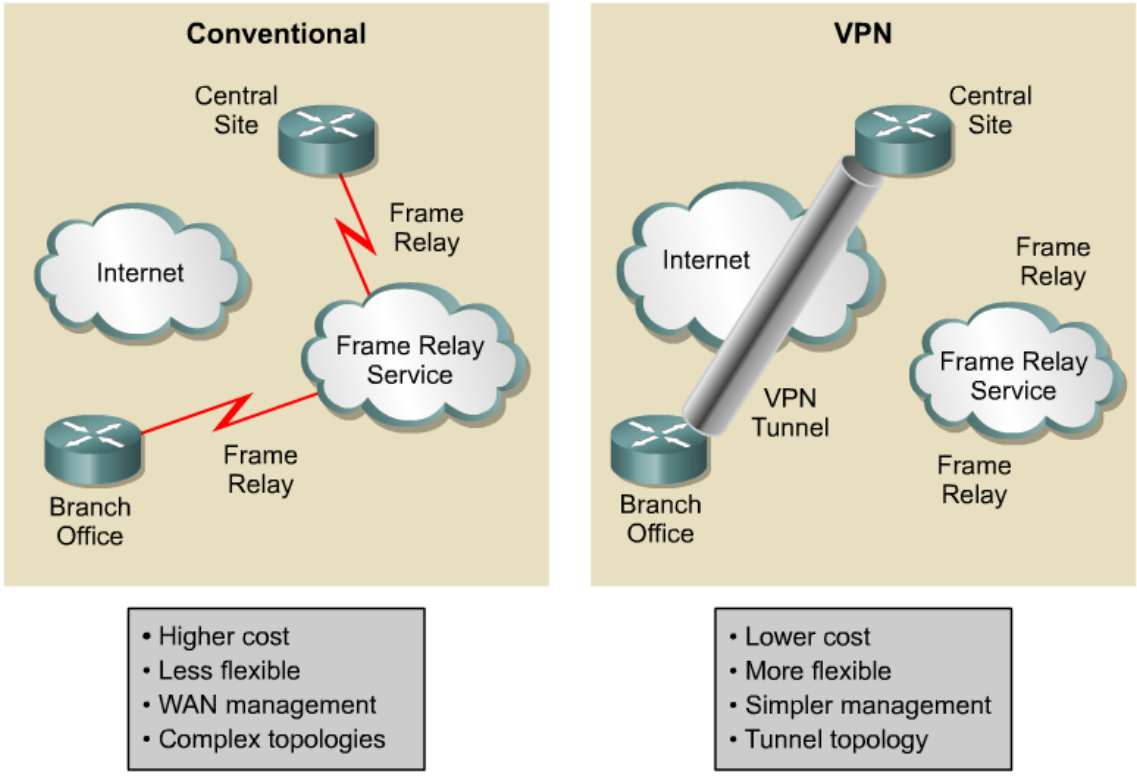
A VPN carries private traffic over a public network using advanced encryption and tunnels to protect:

- Confidentiality of information
- Integrity of data
- Authentication of users

VPNs provide three critical functions:

- **Confidentiality (encryption)** – The sender can encrypt the packets before transmitting them across a network.
  - By doing so, no one can access the communication without permission.
  - If intercepted, the communications cannot be read.
- **Data integrity** – The receiver can verify that the data was transmitted through the Internet without being altered.
- **Origin authentication** – The receiver can authenticate the source of the packet, guaranteeing and certifying the source of the information.

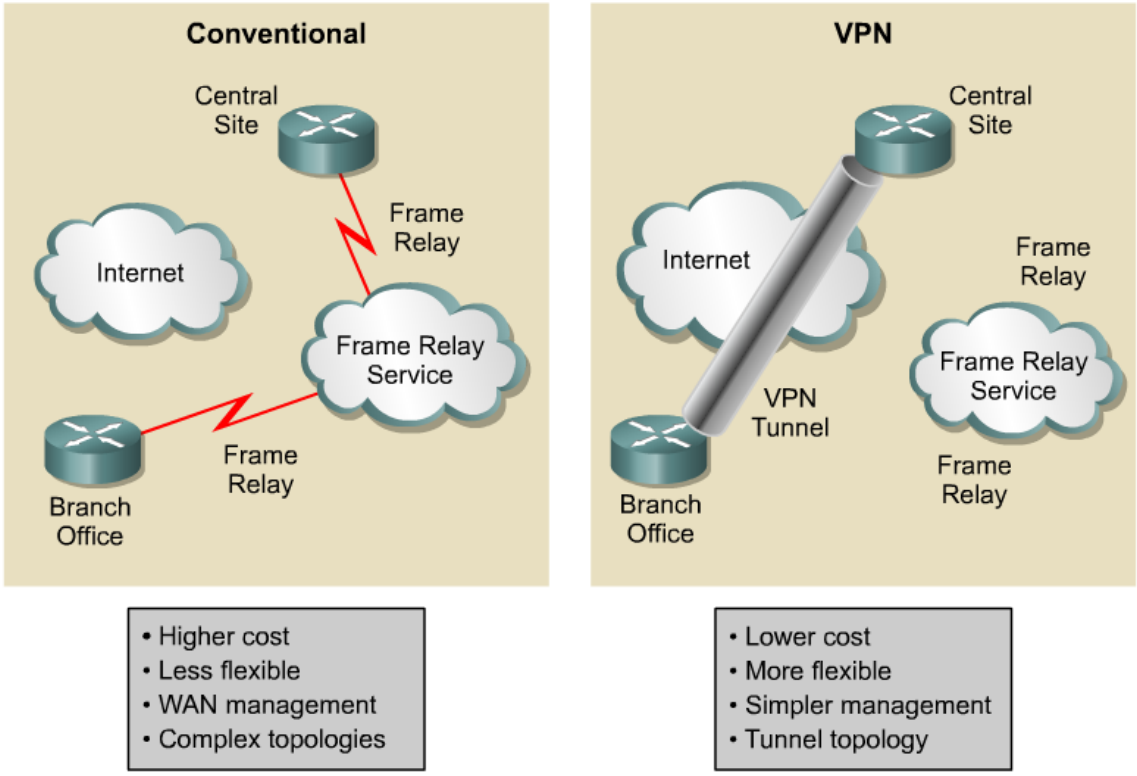
VPN overview



The primary benefits include:

- VPNs offer lower cost than private networks.
  - LAN-to-LAN connectivity costs are typically reduced by 20 to 40 percent over domestic leased-line networks.
- VPNs offer flexibility for enabling the Internet economy.
  - VPNs are inherently more flexible and scalable network architectures than classic WANs.

VPN overview

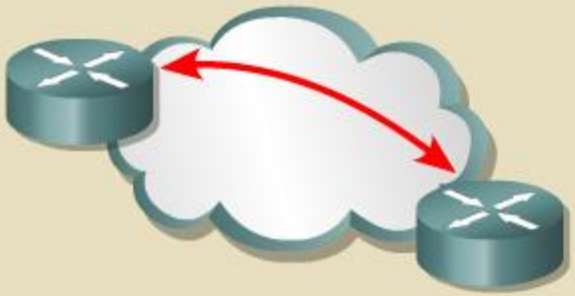


The primary benefits include:

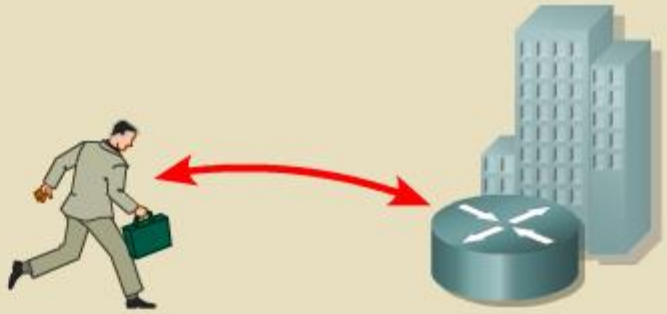
- VPNs offer simplified management burdens compared to owning and operating a private network infrastructure.
- VPNs provide tunneled network topologies that reduce management burdens.
  - An IP backbone eliminates static permanent virtual circuits (PVCs) associated with connection-oriented protocols such as Frame Relay and ATM.

VPN usage scenarios

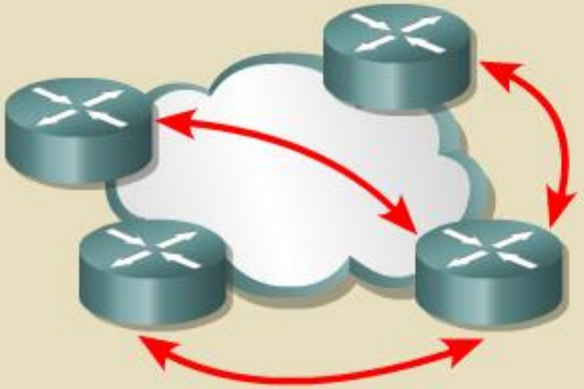
Router to router



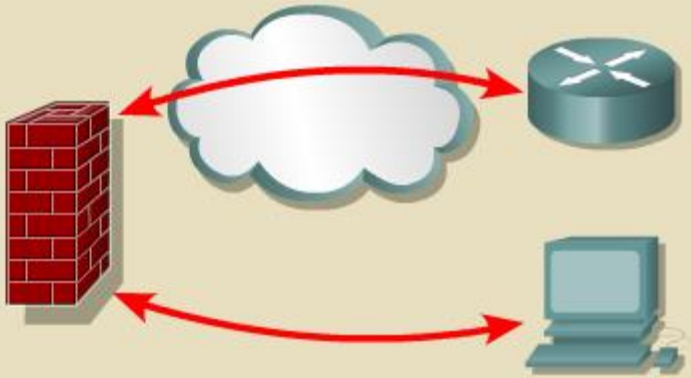
PC to router/concentrator



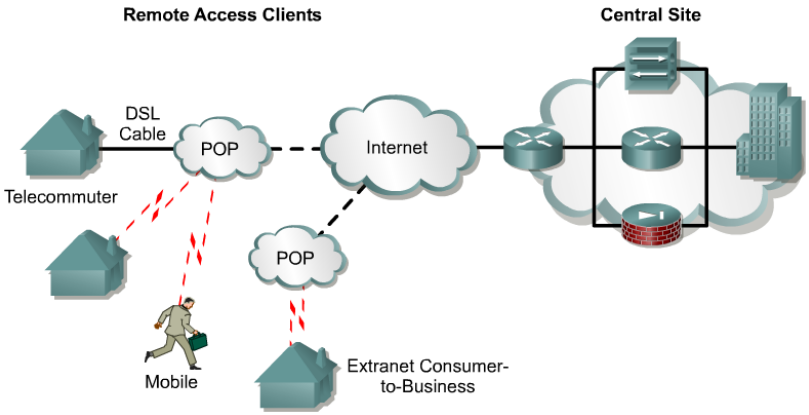
One router to many routers

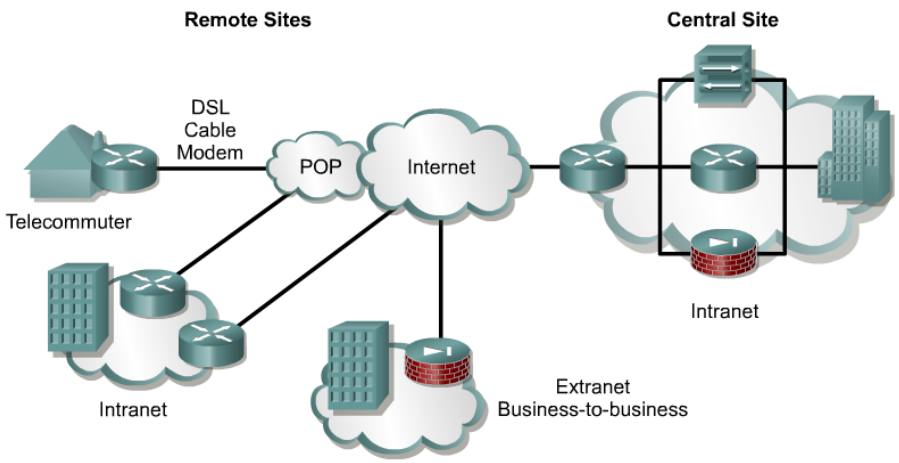


PC to firewall



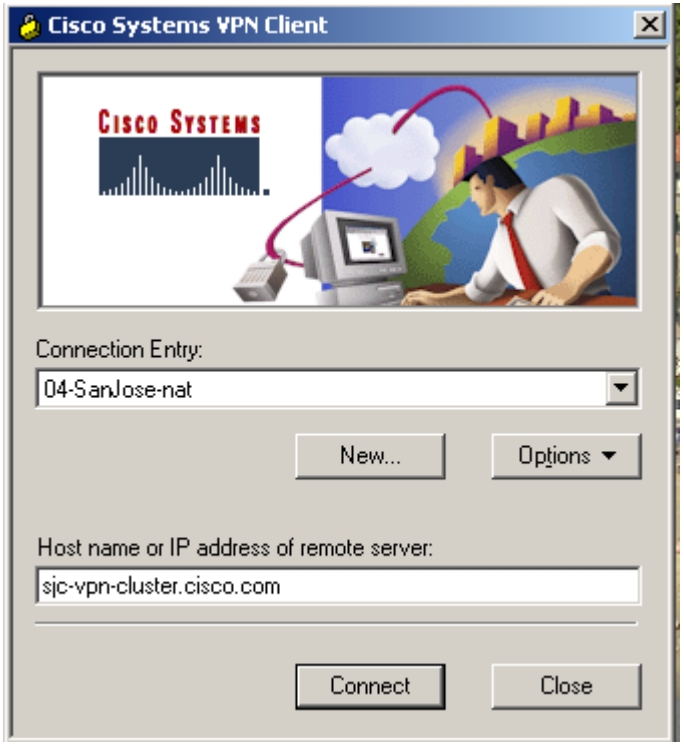
VPN usage scenarios

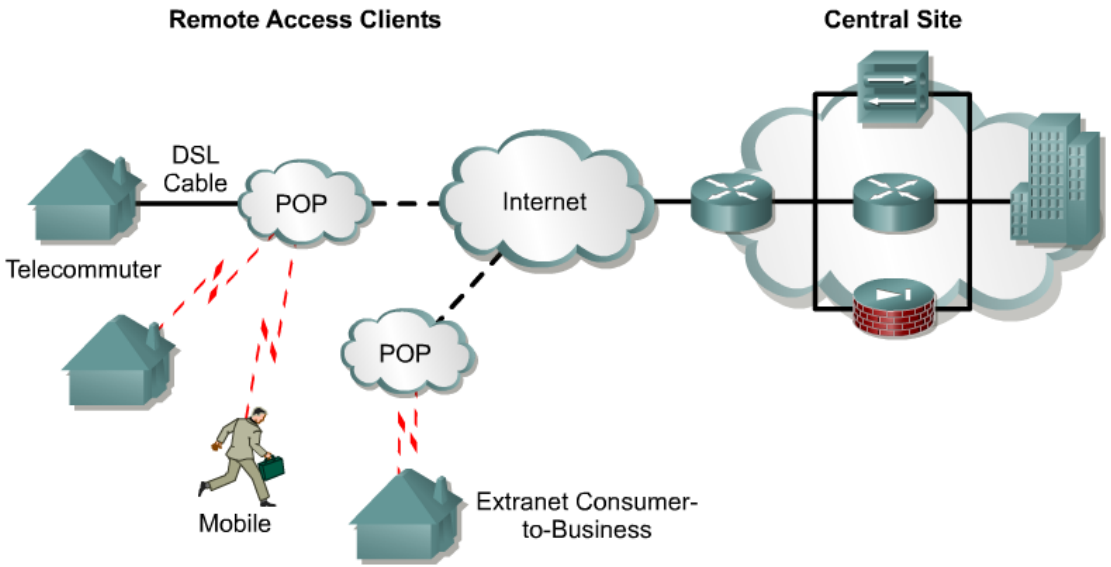




- There are two types of remote access VPNs:
  - **Client-Initiated** – Remote users use clients to establish a secure tunnel across a shared ISP network to the enterprise.
  - **Network Access Server-initiated** – Remote users dial in to an ISP.
    - The NAS establishes a secure tunnel to the enterprise private network that might support multiple remote user-initiated sessions.

Client Initiated VPN





- Remote-access VPNs are an extension of dial networks.
- Remote access VPNs can terminate on head-end devices such as Cisco Routers, PIX Firewalls or VPN Concentrators.
- Remote access clients can include Cisco routers and VPN clients.

VPN Overview

- There are many different approaches to securing your network.
- Application layer scenario
  - Almost any web banking scenario.
  - Access your web banking from any PC in the world.
  - Creates an SSL connection between two applications and transports the data.
  - As long as web browser and web server have same standard implementation of SSL.
  - Disadvantage: Software based encryption which adds processing time and additional CPU cycles.

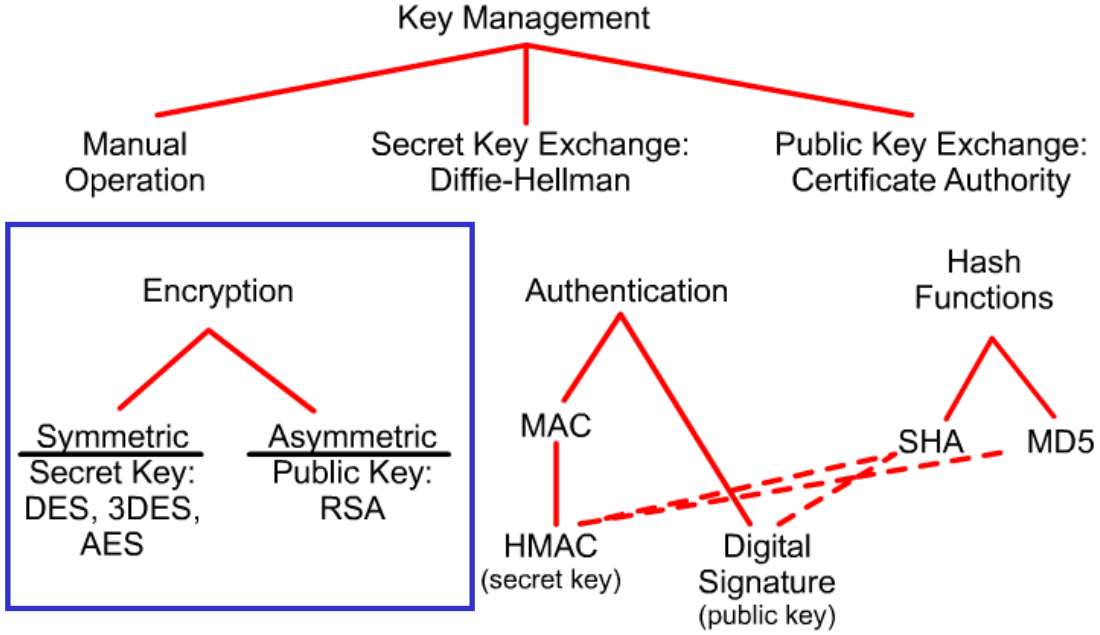
VPN Overview

- Another solution: Data Link Layer encryption
  - Solves the problems of using CPU cycles on the PC.
  - Does not allow you to scale to an ISP-sized environment very easily.
  - Everything from Layer 2 through Layer 7 is encrypted including the network address.
  - Makes it impossible to route the packet until the information is decrypted.
  - Can’t use if crossing any type of public WAN.

VPN Overview

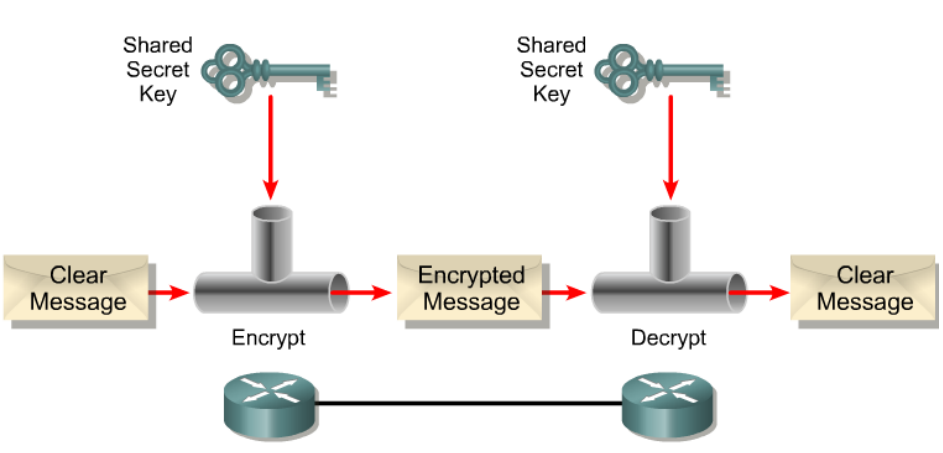
- Another solution: Encryption at network and transport layers.
  - Examples: CET (Cisco Encryption Technology) and IPSec
  - Disadvantage of CET: proprietary (only Cisco equipment)
  - Three necessary components to a good VPN and part of IPSec:
    1. Authentication
    2. Data Integrity
    3. Payload encryption

Encryption Algorithms

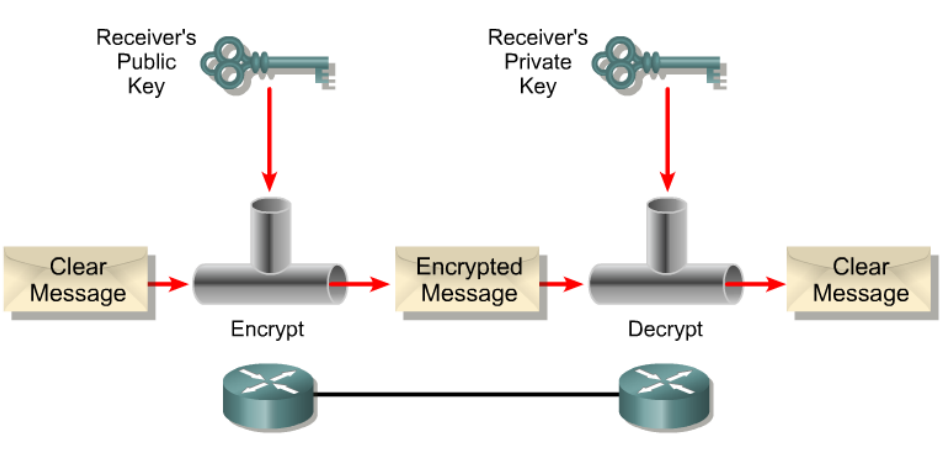


- Some qualities to consider in a good encryption algorithm:
  - Security against cryptographic attacks
  - Scalable, variable length keys
  - Any change to the clear-text input should result in a large change to the encrypted output
  - No restrictions on import of export

Encryption Algorithms



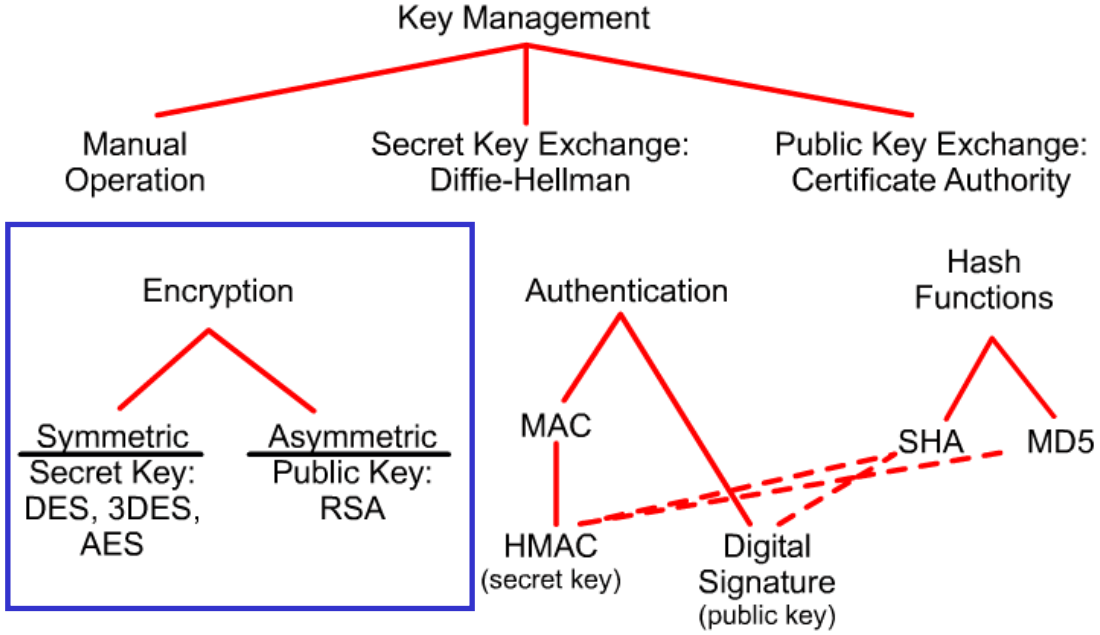
Symmetrical



Asymmetrical

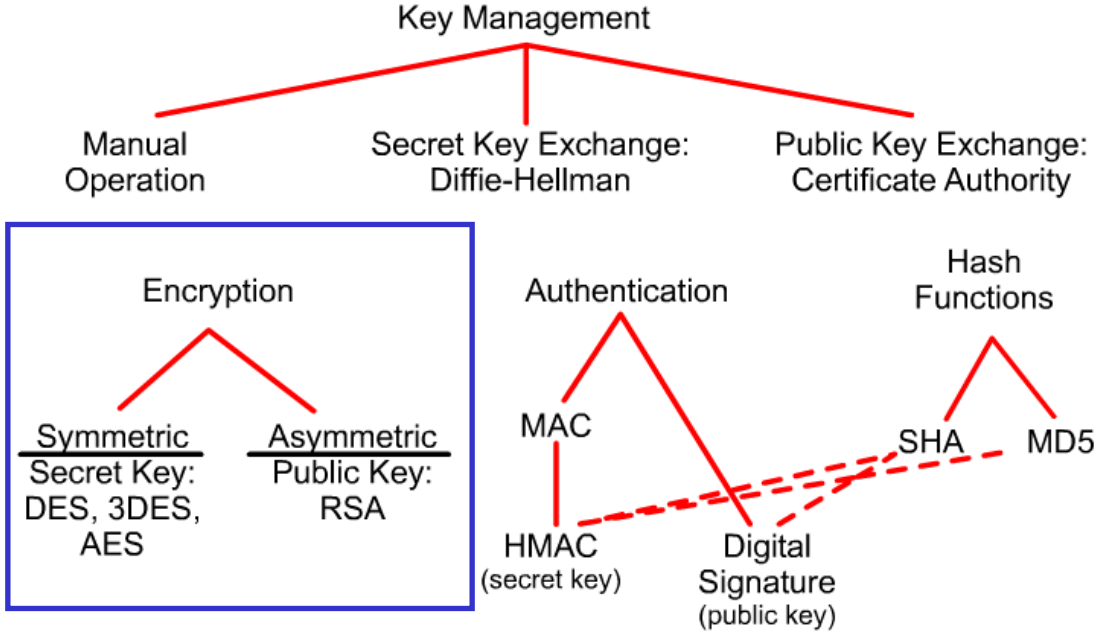
- Symmetrical algorithm – A shared key algorithm that is used to encrypt and decrypt a message.
  - Use the same key to encrypt and decrypt the message.
- Asymmetrical algorithm – Uses a pair of keys to secure encrypt and decrypt a message.
  - Uses one key to encrypt and a different, but related, key to decrypt.

Encryption Algorithms



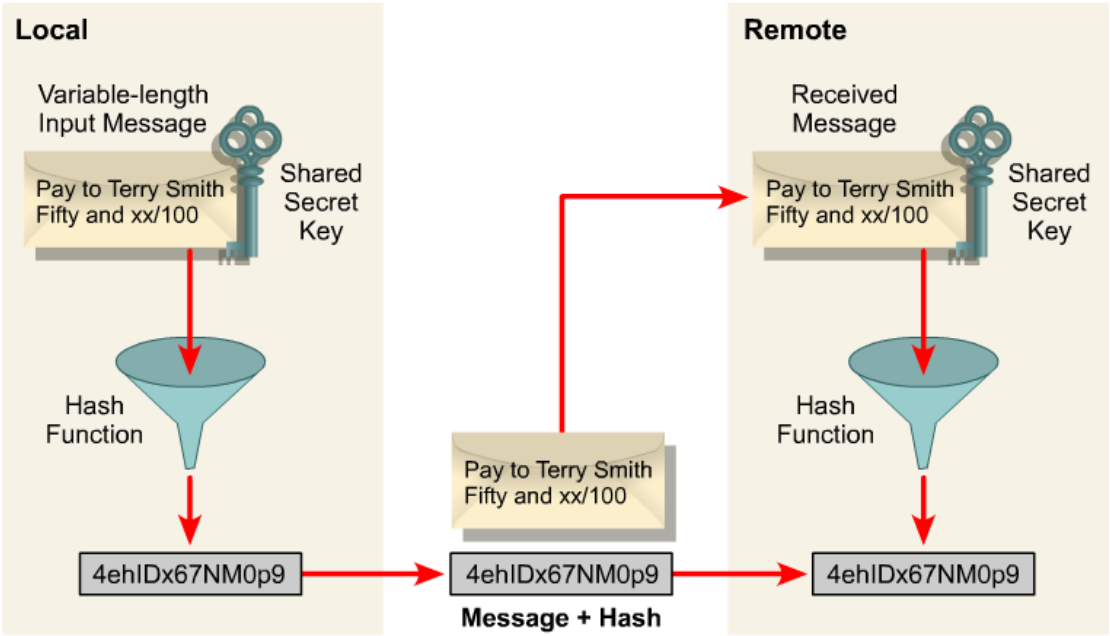
- Common Symmetrical algorithms
  - 56 bit Data Encryption Standard (DES)
  - 168 bit 3DES, “triple DES”
  - 128 or 256 bit Advanced Encryption Standard (AES)
- Advantages of Symmetrical algorithms
  - Speed, fast
  - Mathematical computations are easy to implement in hardware
  - Good for large amounts of data
- Disadvantage of Symmetrical algorithms
  - Sender and receiver share same passwords.
  - There is the problem of how to share the password (key management)

Encryption Algorithms



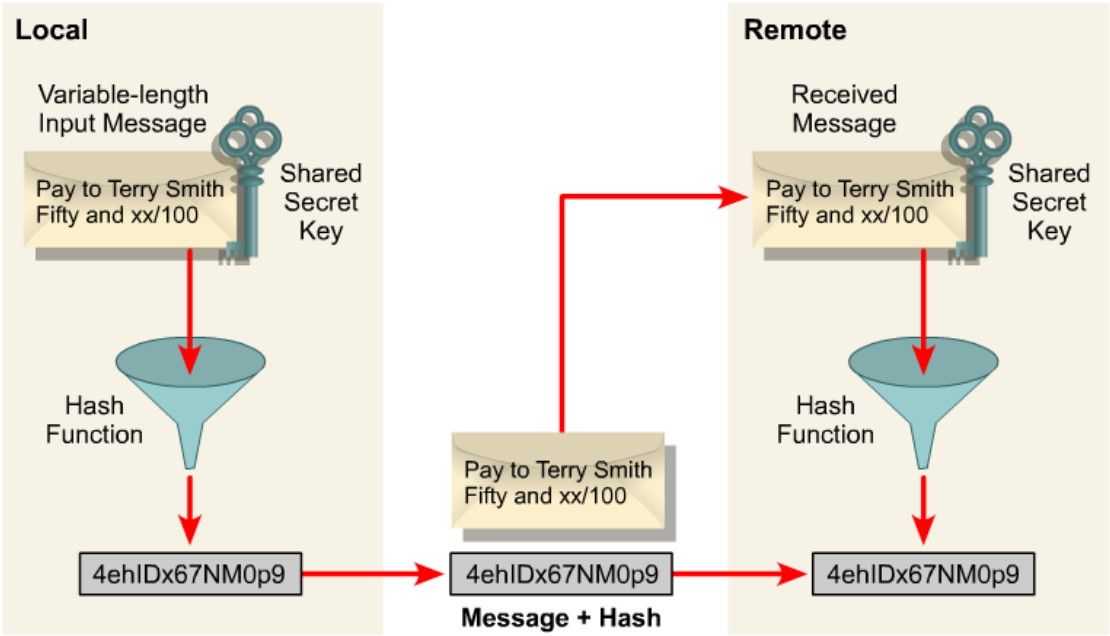
- Common Asymmetrical algorithms
  - RSA, ElGamal, eliptic curves
- Advantages of Symmetrical algorithms
  - No problems with key management, one key is kept private and the other key is public and given to anyone that needs to encrypt data.
  - Great for authentication because you are the only one with the private key used to decrypt the data.
  - Can be used for digital signatures, authenticated key exchanges, email or small amounts of data.
  - Based on very hard mathematical equations.
- Disadvantage of Symmetrical algorithms
  - Slower in encrypting than symmetrical algorithms

Hashing



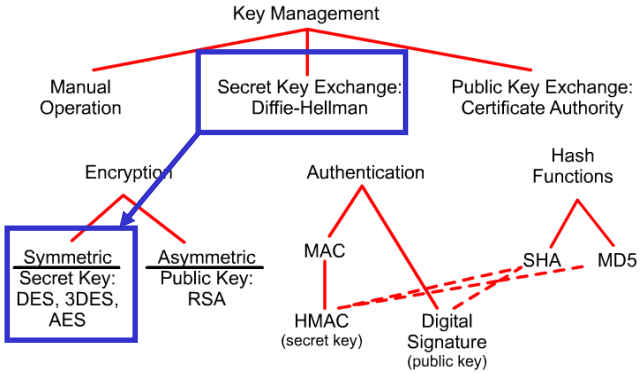
- Hashing is used for data integrity.
- Hashing algorithms is a one-way algorithm that produces a fixed-length output, no matter what the size of the input is.
- Analogy:
  - Blender with 3 small oranges and 3 big oranges
  - Blend it and make one cup of juice
  - Your neighbor can do the exact same thing
  - You can never reverse-engineer the output to get the input.
  - You can’t determine that 3 big and 3 small oranges were used to make the one glass of juice.

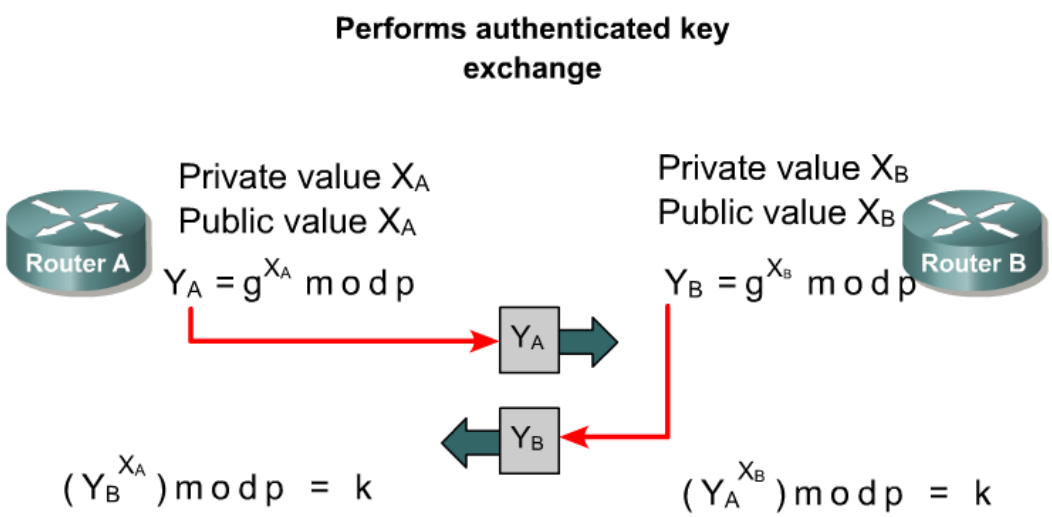
Hashing



- Two common hashing algorithms:
  - MD5: fixed-length 128 bit output
  - SHA-1: fixed-length 160 bit output (preferred, less likely to result in a collision (two different inputs giving the same output).
- Qualities in a good hashing algorithm:
  - High resistance to cryptographic attack
  - Any change to the clear-text input results in a large change in the encrypted output.
  - The probability of collision is low.

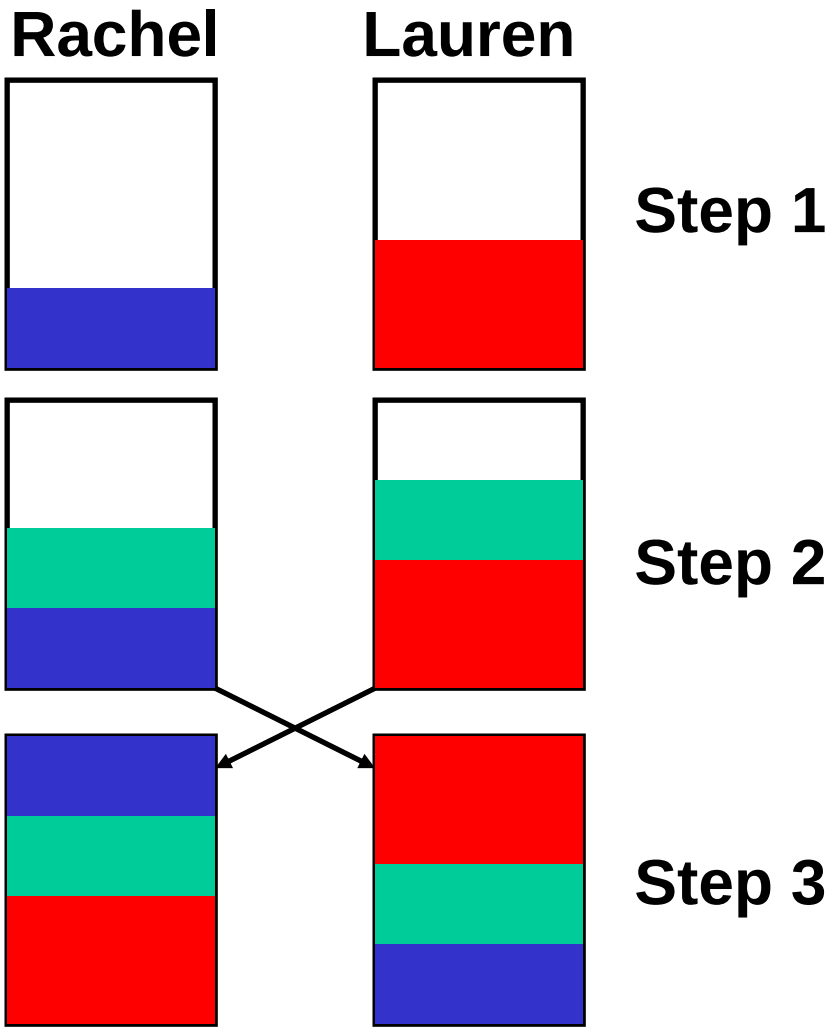
Diffie-Hellman algorithm





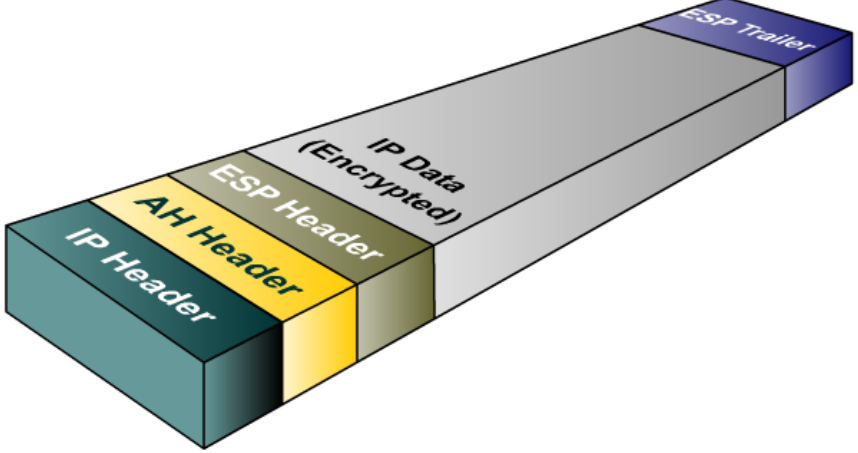
- In a VPN network, fast, strong encryption is a must.
- This is why most implementations use a symmetrical algorithm to do payload encryption.
- Problem with symmetrical algorithms is key management.
- Diffie-Hellman helps solve this.
- Used for automatic secure key exchange of symmetrical “shared” keys (and other types of keys) across an insecure network for IPSec.

Diffie-Hellman algorithm - simplified



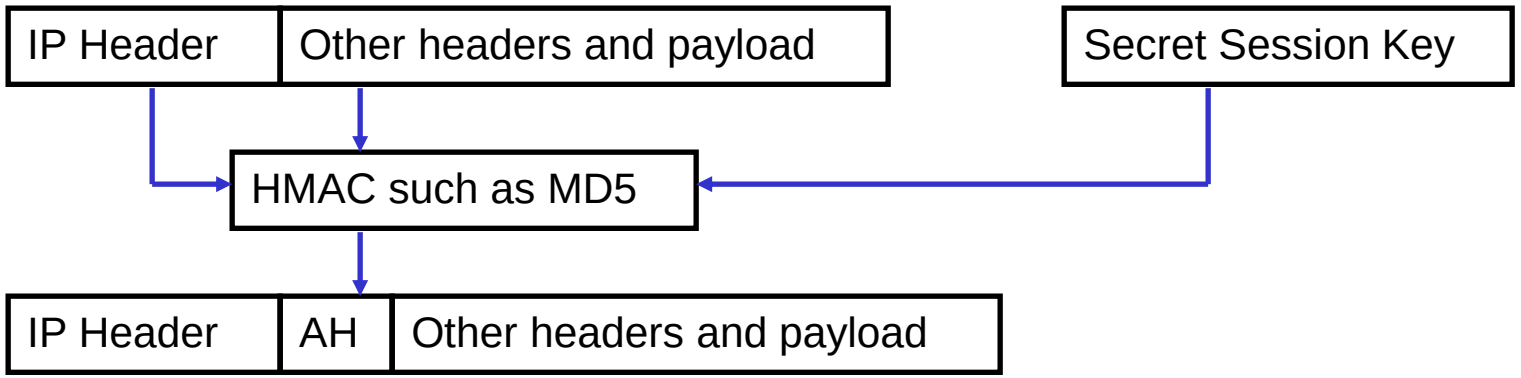
- Step 1: Rachel chooses her secret color and adds 1 liter of blue to her paint can. Lauren chooses her secret color red and adds 1.5 liters of red.
- Step 2: A shared secret color is chosen that both Rachel and Lauren add 1.1 liters of green to their can.
- Step 3:The paint cans are exchanged. Rachel adds 1 liter of her original secret color blue to the paint can she got from Lauren. Lauren adds her 1.5 liters original secret color red to the paint can she got from Rachel.
- Now both paint cans have the identical colors with the same amounts.

IPSec Overview



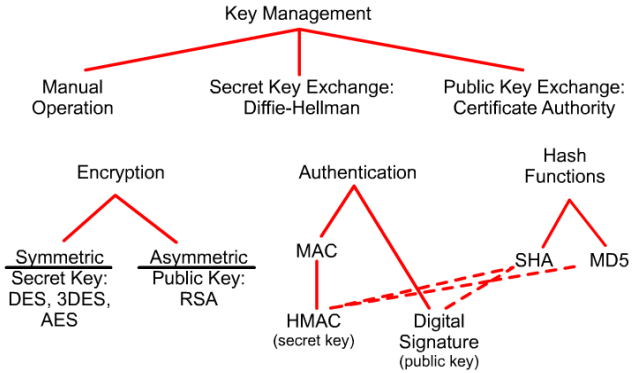
- IPSec was designed to work at Layers 3 and 4.
- Using different options can:
  - Authenticate
  - Check for data integrity
  - Encrypt the payload portion of IP
- IPSec can be used between:
  - Two gateways
  - Two hosts
  - Host and its gateway
- Two primary protocols:
  - Authentication Header (AH)
  - Encapsulation Security Protocol (ESP)

AH – Authentication Header



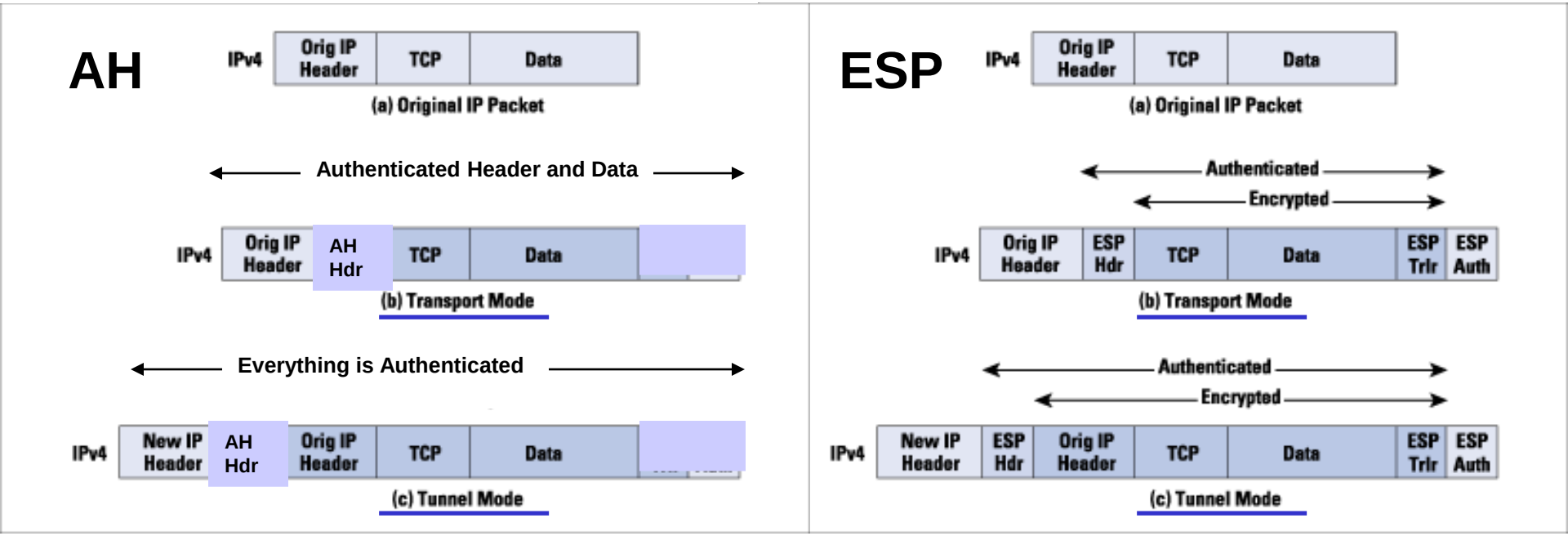
- AH provides:
  - data integrity
  - authentication
- Does not provide encryption
- Uses one-way has function (also called an HMAC) to guarantee data integrity and origin of the packet.
- Entire IP packet put through one-way hash.
- Includes IP header which could lead to problems.
- TTL must be “zeroized: to give a “standard header”
- Produces a new AH header for the packet to be transmitted.
- AH may be applied alone, in combination with the IP ESP.

ESP – Encapsulating Security Protocol



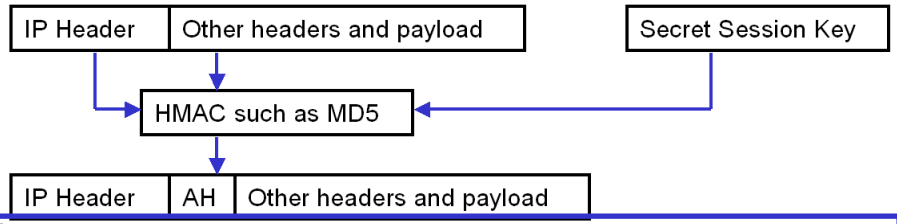
- ESP is primarily used to provide payload encryption.
- With current revisions of the RFC, it also includes the ability for authentication and integrity.
- Because ESP can include all three services, authentication, integrity, and encryption, most implementations do not include an AH options.
- IPSec can use different algorithms for payload encryption such as:
  - DES
  - 3DES
  - AES

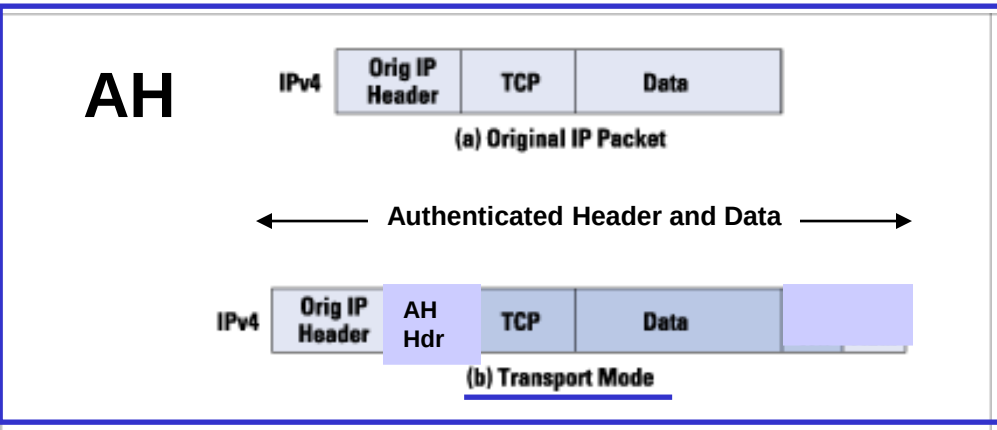
Tunnel Mode versus Transport Mode

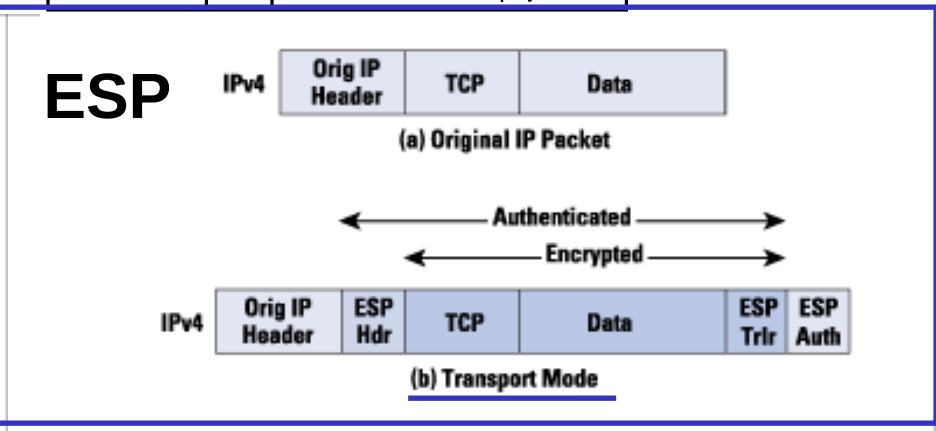


- Both AH and ESP can operate in two modes:
  - Transport Mode
  - Tunnel Mode (default)
- **Transport Mode** – The original IP packet is put through the ESP and/or AH options and then the original IP header is reused with the packet, which would be the original packet plus added information from ESP and/or AH.
- **Tunnel mode** – The original IP packet is put through the ESP and/or AH options and the a new IP header is created for the new packet, which is a combination of the original packet plus ESP and/or AH information plus a new IP header.

Transport Mode

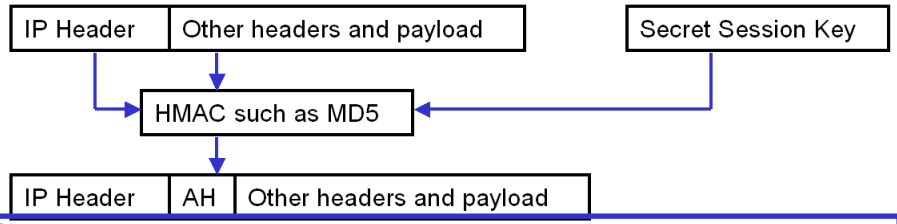




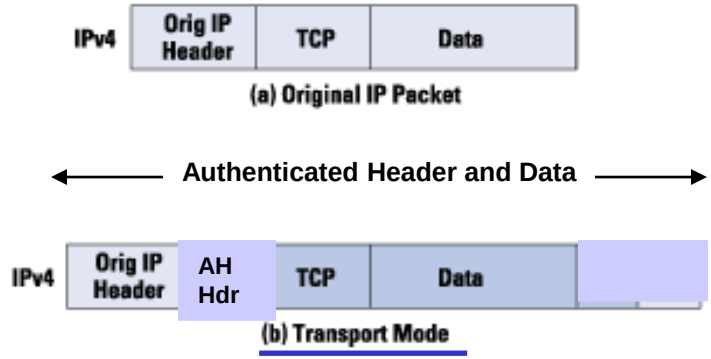


- Transport mode
  - Current IP header has been used in the hashing algorithm and therefore cannot be changed from sender to receiver.
  - If the packet goes through any device that performs NAT/PAT, then a portion of the IP header is changed and you will never get the same hash output, because of different inputs at the sender and receiver ends.
  - Therefore, the packet will never be validated at the receiving end.

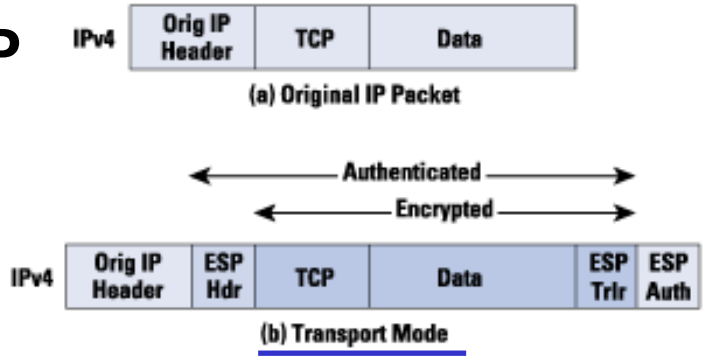
Transport Mode



AH

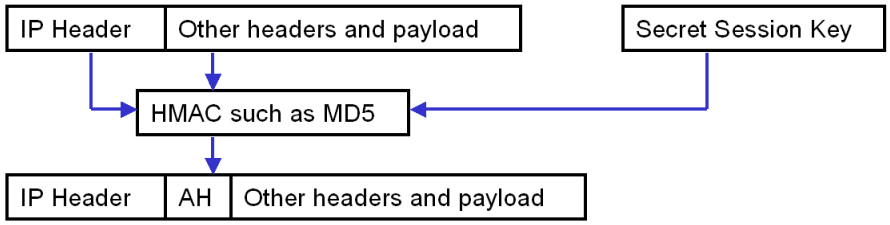


ESP

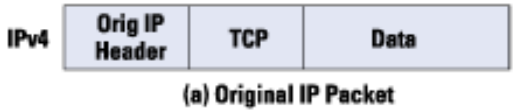


- Transport mode
  - Should only be used if:
    - You have control of the network from end to end
    - Guarantee no IP packet manipulation will take place.

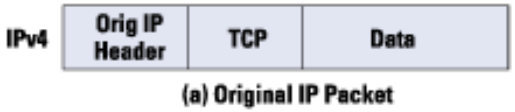
Tunnel Mode

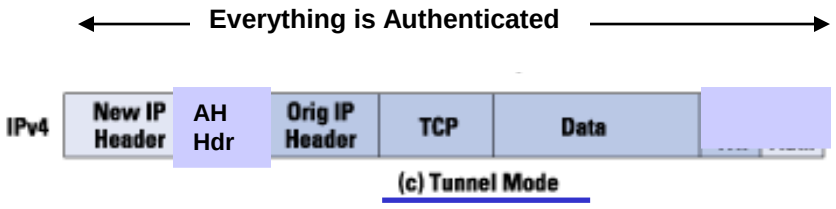


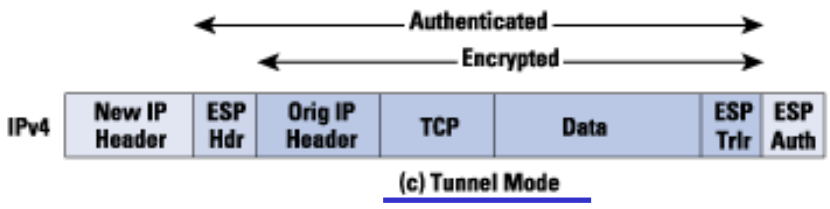
AH



ESP

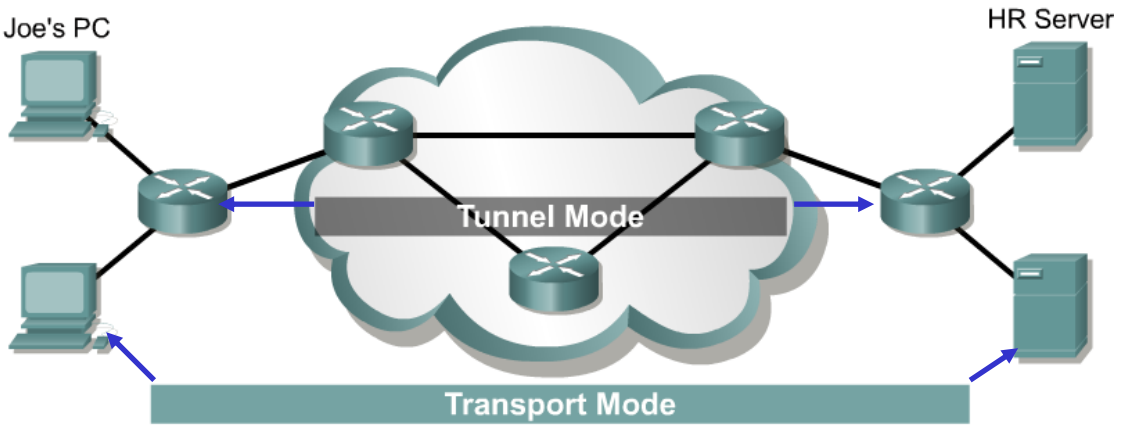






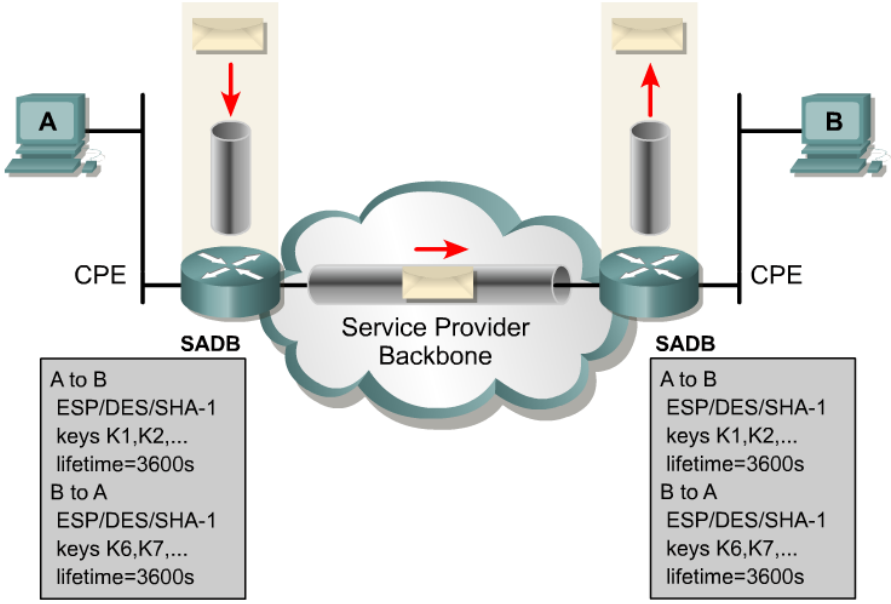
- Tunnel mode
  - A new IP header is used from gateway device to gateway device, and the original packet is tunneled inside.
  - Once the receiving end receives the packet:
    - Removes the new IP header
    - Decrypts original header
    - A new tunnel header can be added, which can get manipulated (NAT) throughout the network without affecting the tunneled protocol.

Tunnel Mode versus Transport Mode



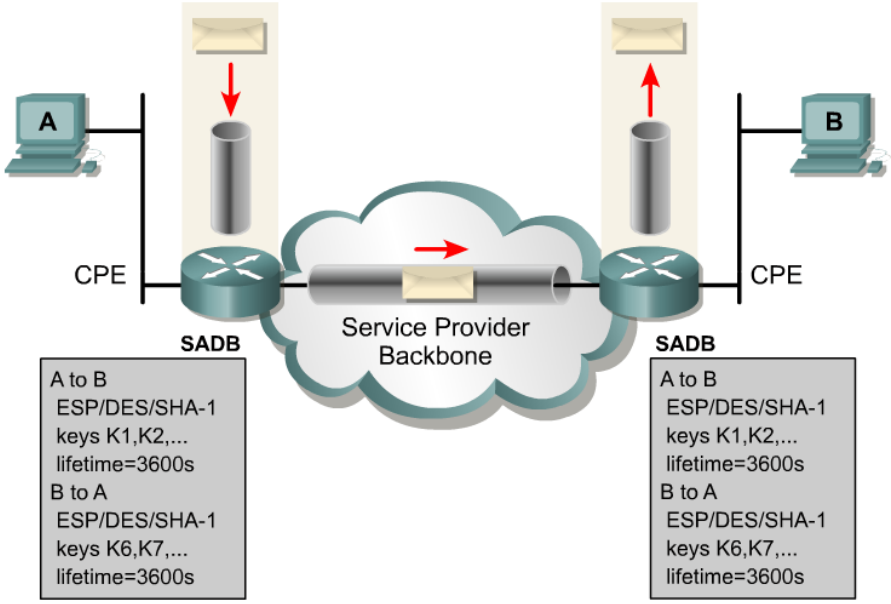
- In **transport mode end hosts do IPSec** encapsulation of their own data (host-to-host) therefore IPSec has to be implemented on each of the end-hosts.
  - The application endpoint must be also the IPSec endpoint.
  - ESP transport mode is **used between hosts.**
- In **tunnel mode IPSec gateways provide IPSec** services to other hosts in peer-to-peer tunnels, and end-hosts are not aware of IPSec being used to protect their traffic.

SA - Security Associations



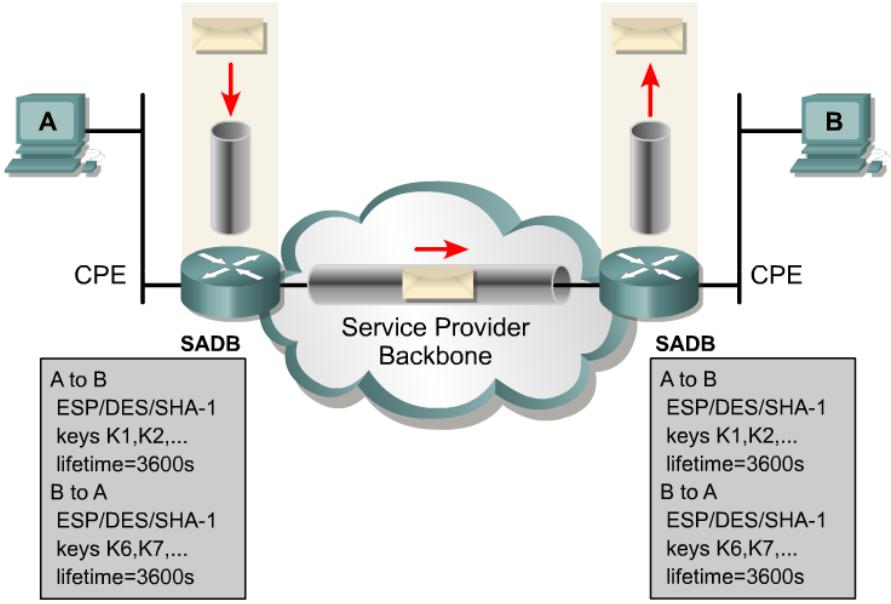
- Before an IPSec tunnel/transport can be created, certain parameters must be negotiated and kept track of.
- Security Associations (SAs) represent a **policy contract between two peers or hosts**, and describe how the peers will use IPSec security services to protect network traffic.
- SAs contain all the security parameters needed to securely transport packets between the peers or hosts, and define the security policy used in IPSec.
- Every VPN device has to have some form of security policy database (SPD), referred to as a Security Associate (SA).
- VPN devices store all their active SAs in a local database called the SA database (SADB).

SA - Security Associations



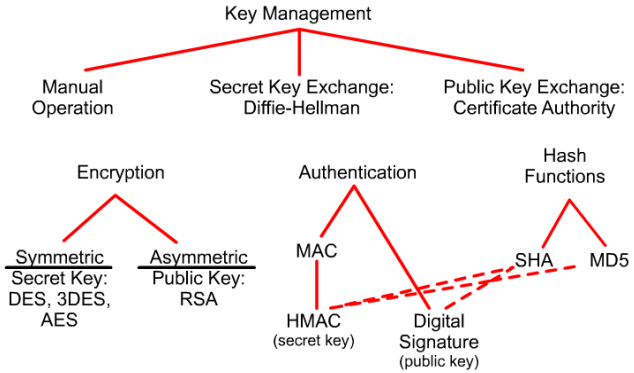
- An SA is a single connection and all the parameters associated with it that are agree upon by the two devices participating in the exchange.
- Each SA is unidirectional.
- There will always be at least two SAs in your SPD, one for A to B and B to A.
- Possible to have multiple peers in a VPN network (NAS).
- Each SA gets a unique 32 bit Security Parameter Index (SPI) number that is sent in every packet pertaining to the specific SA.

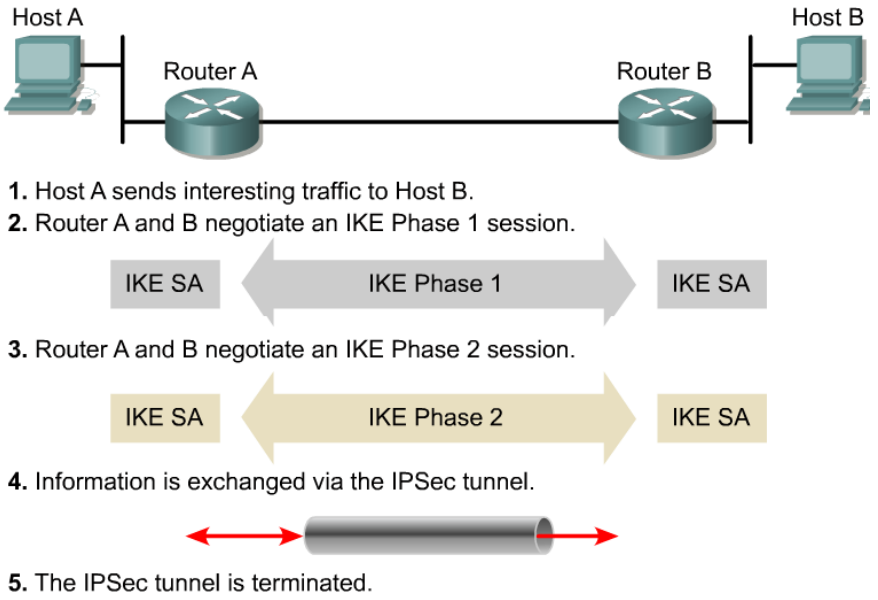
SA - Security Associations



- The SA keeps track of general information such as :
  - source IP
  - destination IP
  - IPSec protocols used
  - SPI, encryption and authentication algorithms
  - key lifetime (sets the amount of time and/or byte count that a key is valid for; longer the time, the more vulnerable the data is.)

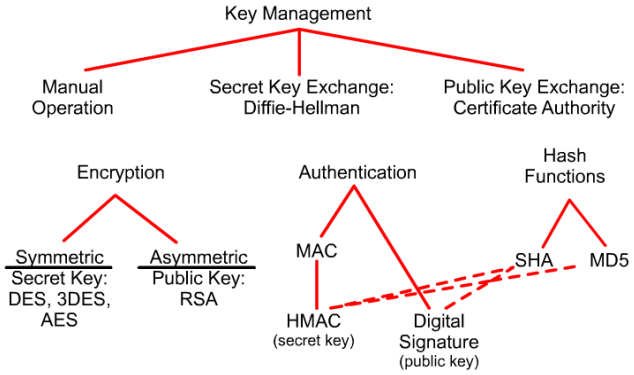
IKE – Internet Key Exchange

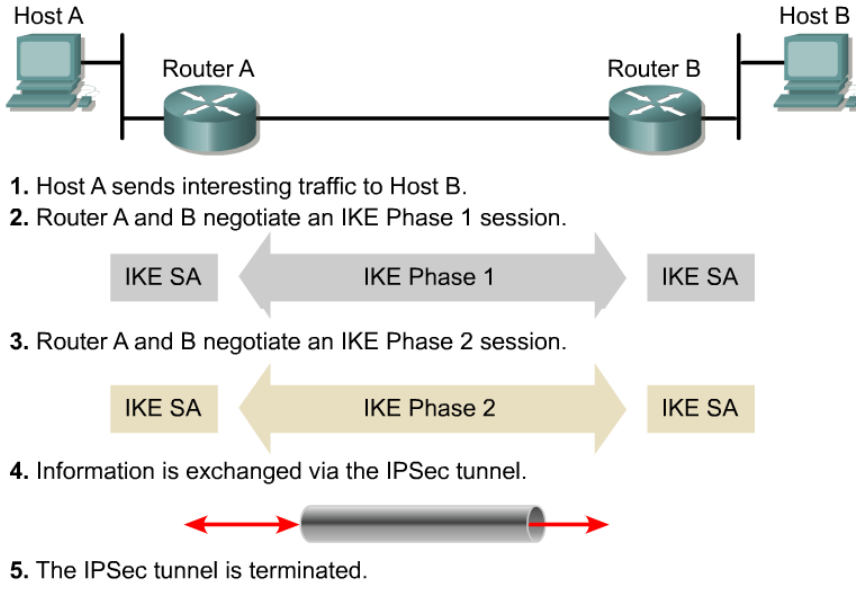




- Internet Key Exchange (IKE) is used to establish all the information needed for a VPN tunnel.
- Within IKE:
  - Security policies are negotiated
  - SAs are established
  - Create and exchange keys that will be used by other algorithms such as DES
- There are two phases to IKE…

IKE – Phase One

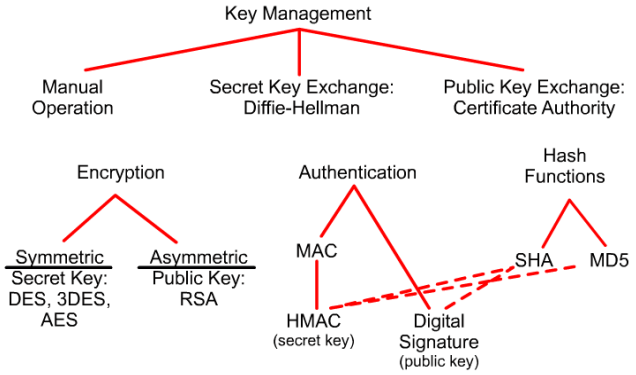


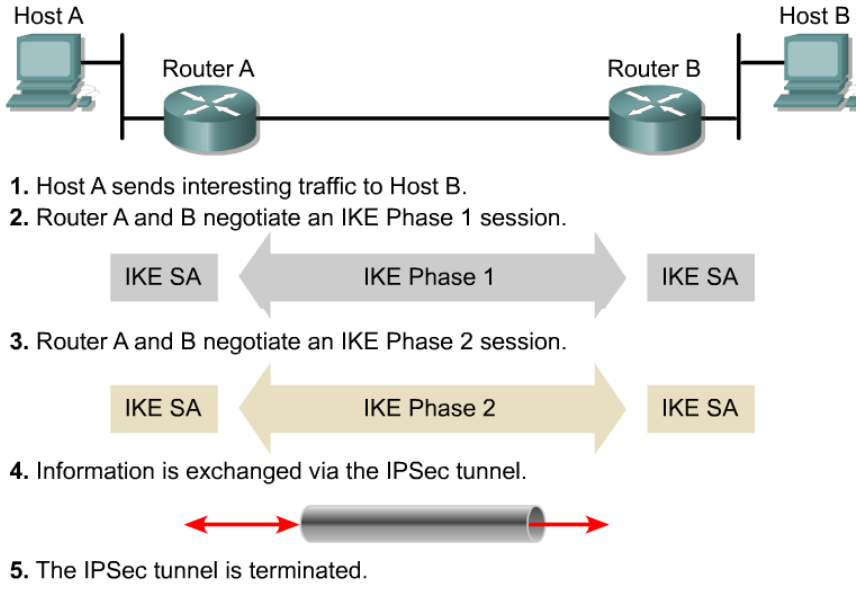


- Phase One
  - Used to negotiate policy sets
  - Authenticate peers
  - Create a secure channel between peers
- Standard policy set:

Parameter	Strong	Stronger
Encryption algorithm	DES	3DES
Hash algorithm	MD5	SHA-1
Authentication method	Preshared	RSA signatures
Key exchange	Diffie-Hellman group 1	Diffie-Hellman group 2
IKE SA lifetime	86,400 seconds	Less than 86,400 secs.

IKE – Phase One

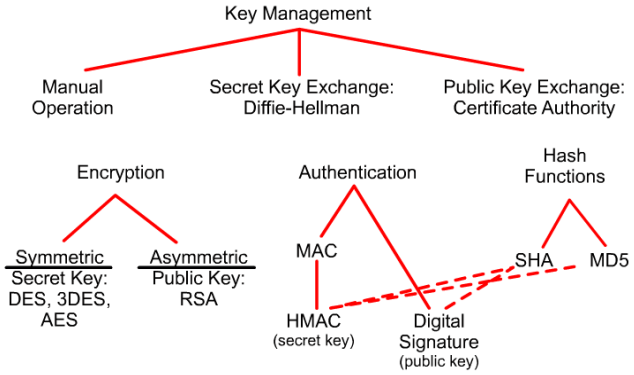


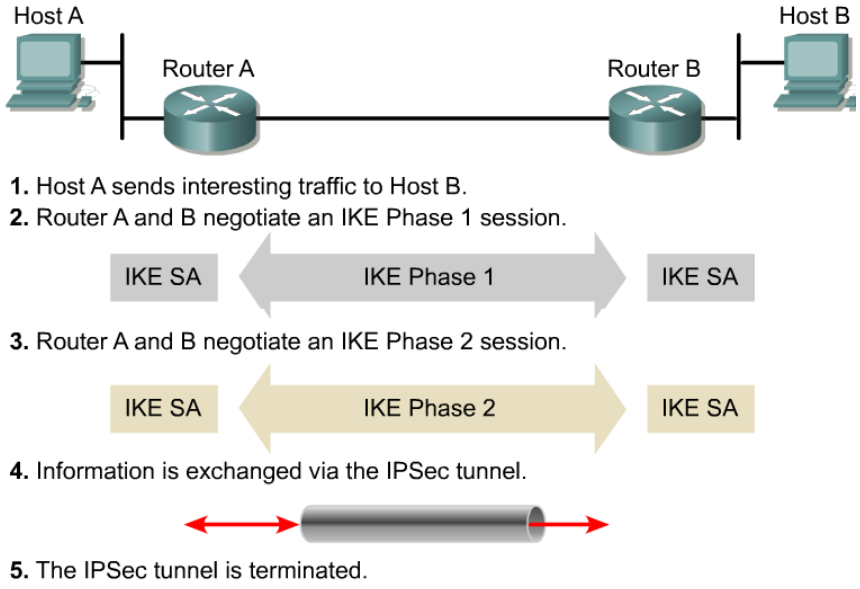


- Phase One
  - Main mode: three different and distinct exchanges take place to add to the security of the tunnel.
  - Aggressive mode: Everything is sent in a single exchange.

Parameter	Strong	Stronger
Encryption algorithm	DES	3DES
Hash algorithm	MD5	SHA-1
Authentication method	Preshared	RSA signatures
Key exchange	Diffie-Hellman group 1	Diffie-Hellman group 2
IKE SA lifetime	86,400 seconds	Less than 86,400 secs.

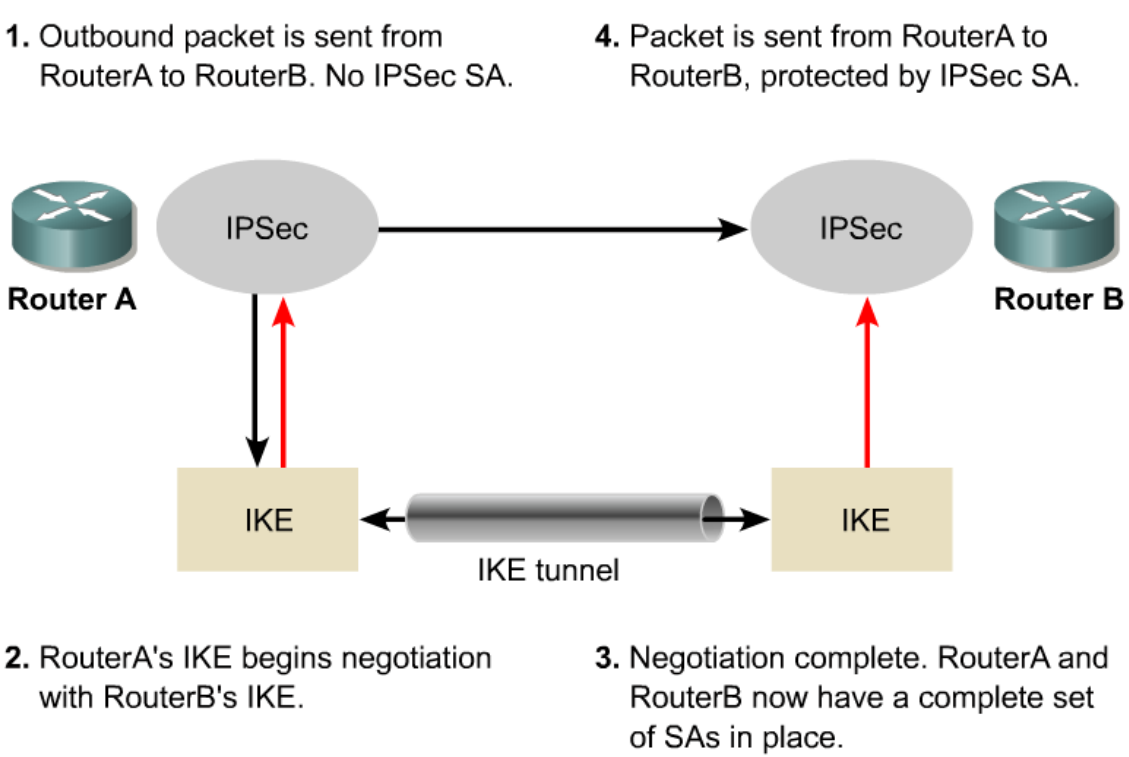
IKE – Phase Two





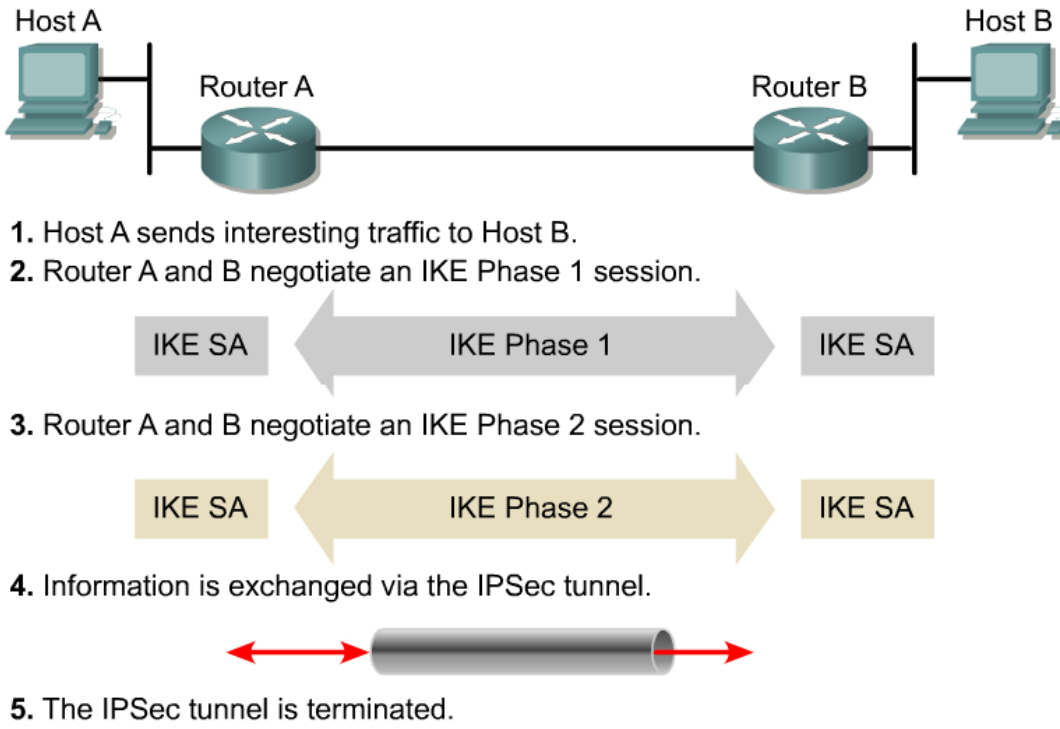
- Phase Two
  - Used to:
    - Negotiate the IPSec security parameters
    - Establish SAs
    - Optionally perform Diffie-Hellman Key exchanges
  - Has one mode, quick mode, which happens after Phase One.

IKE protects SAs



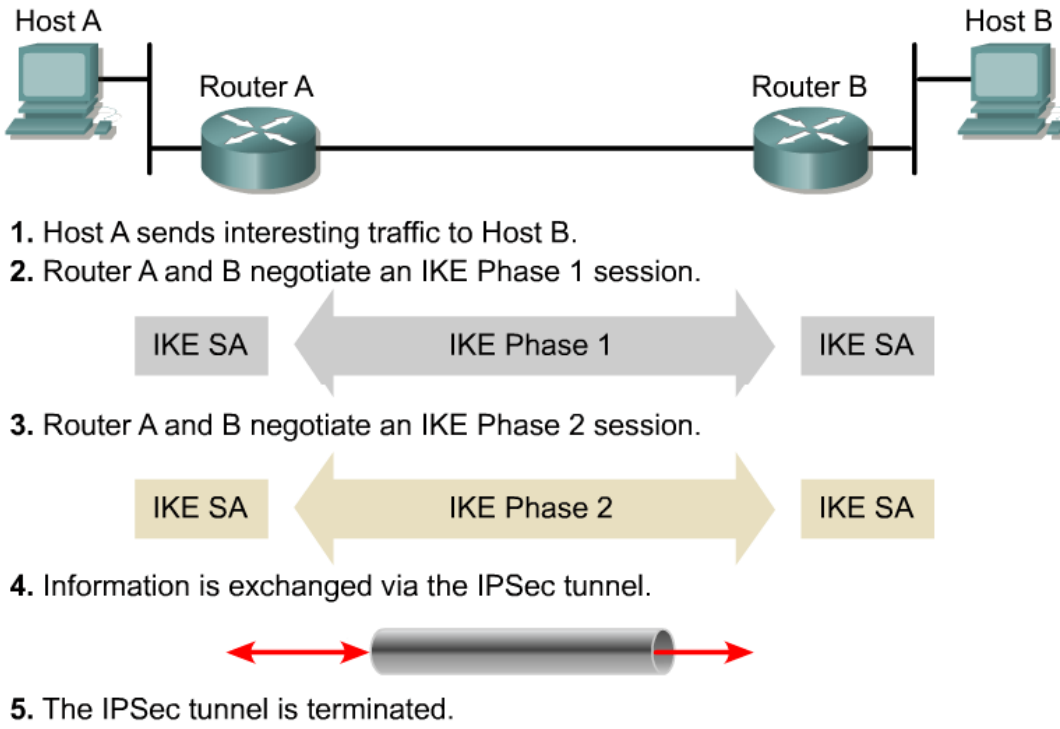
- The IKE tunnel protects the SA negotiations.
- After the SAs are in place, IPSec protects the data that A and B exchange.
- IKE Mode configuration allows a gateway to download an IP address (and other network-level configuration) to the client as part of an IKE negotiation.

Preparing for IKE and IPSec



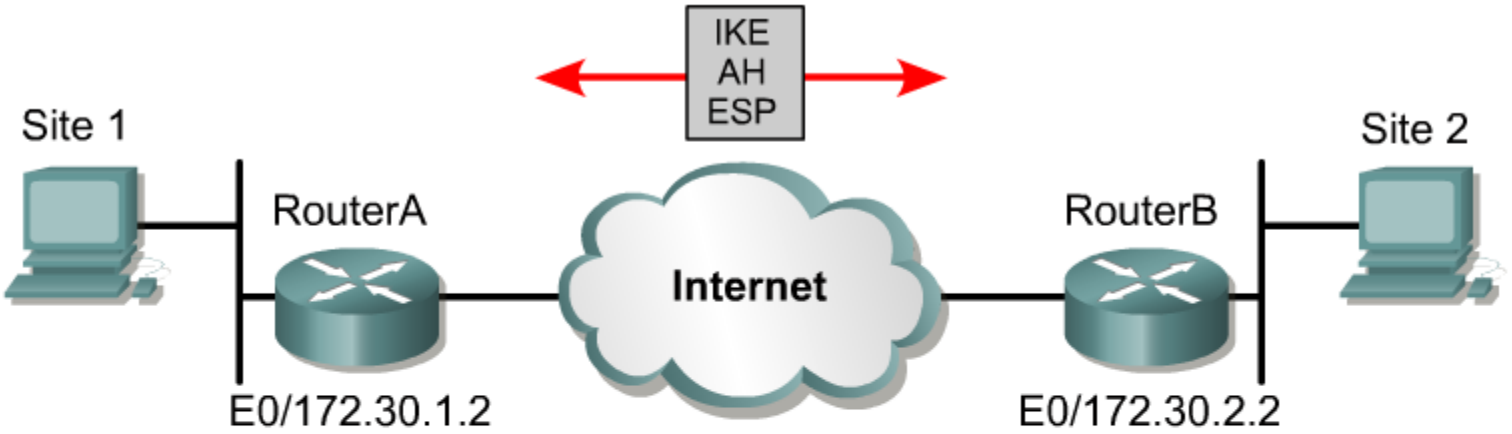
- Step 1 – Define interesting traffic that should be protected.
- Step 2 – Perform IKE phase 1 – negotiate the security policy, etc.
- Step 3 – Perform IKE phase 2 – negotiate SAs, etc.
- Step 4 – Transfer data – encrypt interesting traffic and send it to peer devices,.
- Step 5 – Tear down the tunnel.

Preparing for IKE and IPSec



- Do my current ACLs allow ESP, AH, and IKE to terminate on the router?
- What interesting traffic needs to be encrypted?
- What phase one policies can I support?
- What phase two policies will be implemented?
- Does the network route properly before I add encryption services? (ping?)

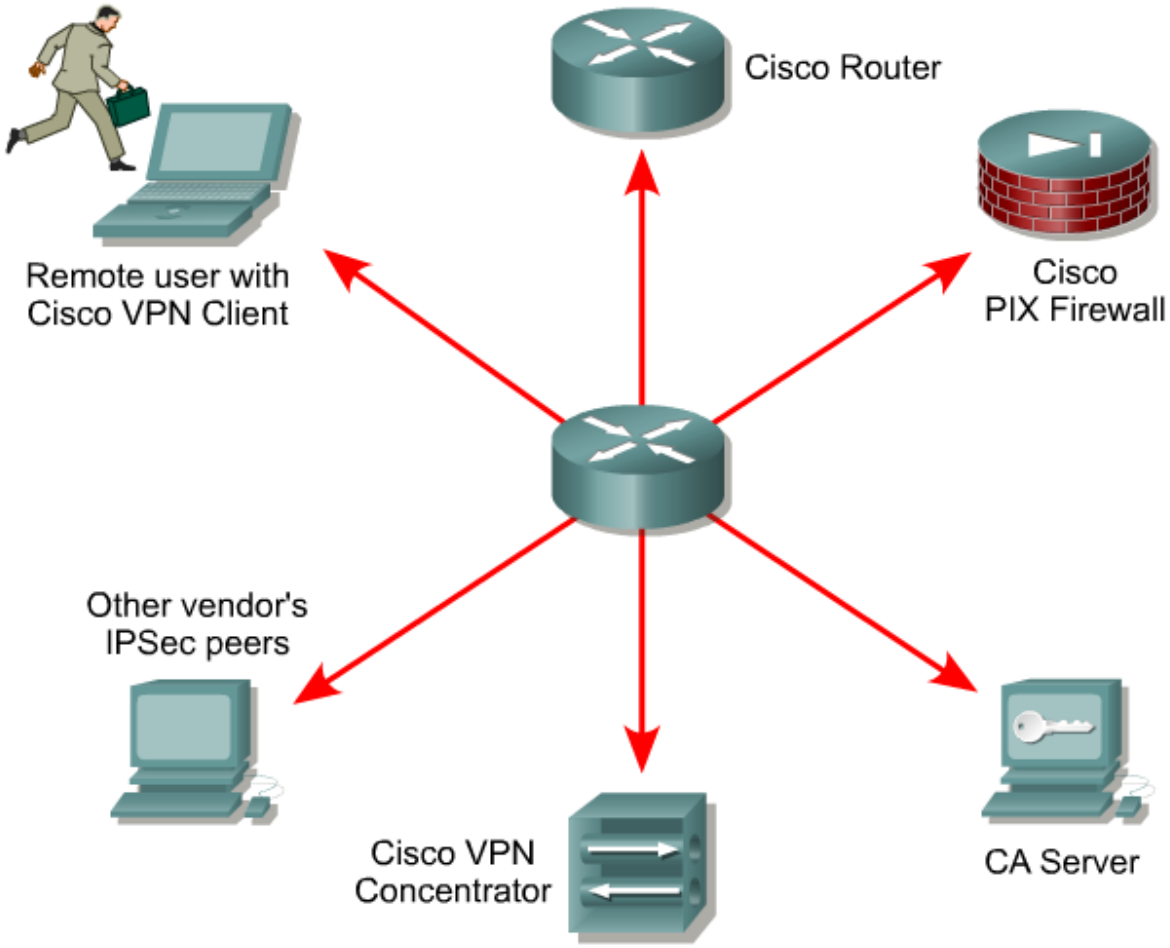
Do my current ACLs allow ESP, AH, and IKE to terminate on the router?




```
RouterA#show access - lists
access - list 102 permit ahp host 172.30.2.2 host 172.30.1.2
access - list 102 permit esp host 172.30.2.2 host 172.30.1.2
access - list 102 permit udp host 172.30.2.2 host 172.30.1.2
   eq lsakmp
```

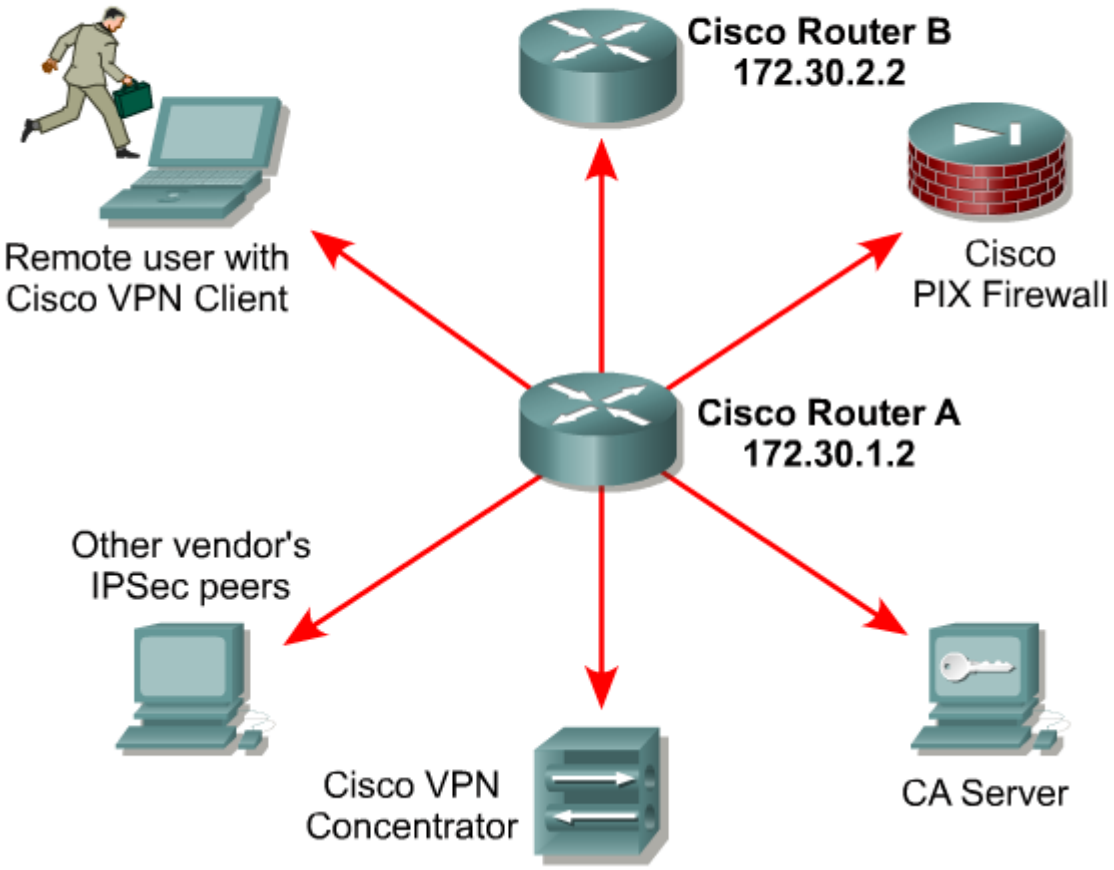
Ensure protocols 50 and 51, and UDP port 500 traffic are not blocked at interfaces used by IPSec.

Identify IPSec peers

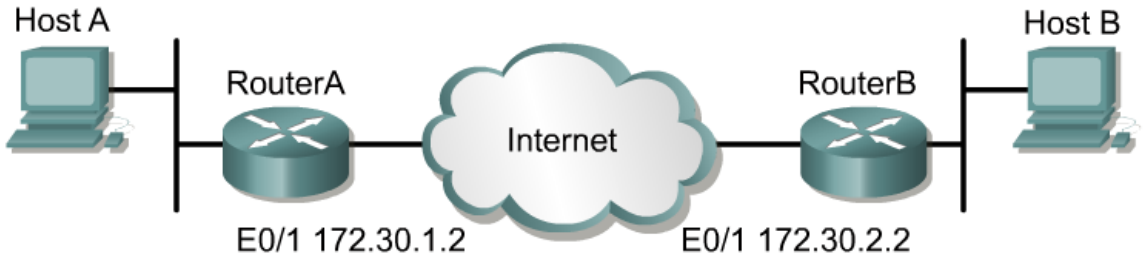


Ensure the network works

```
RouterA#ping 172.30.1.2
```



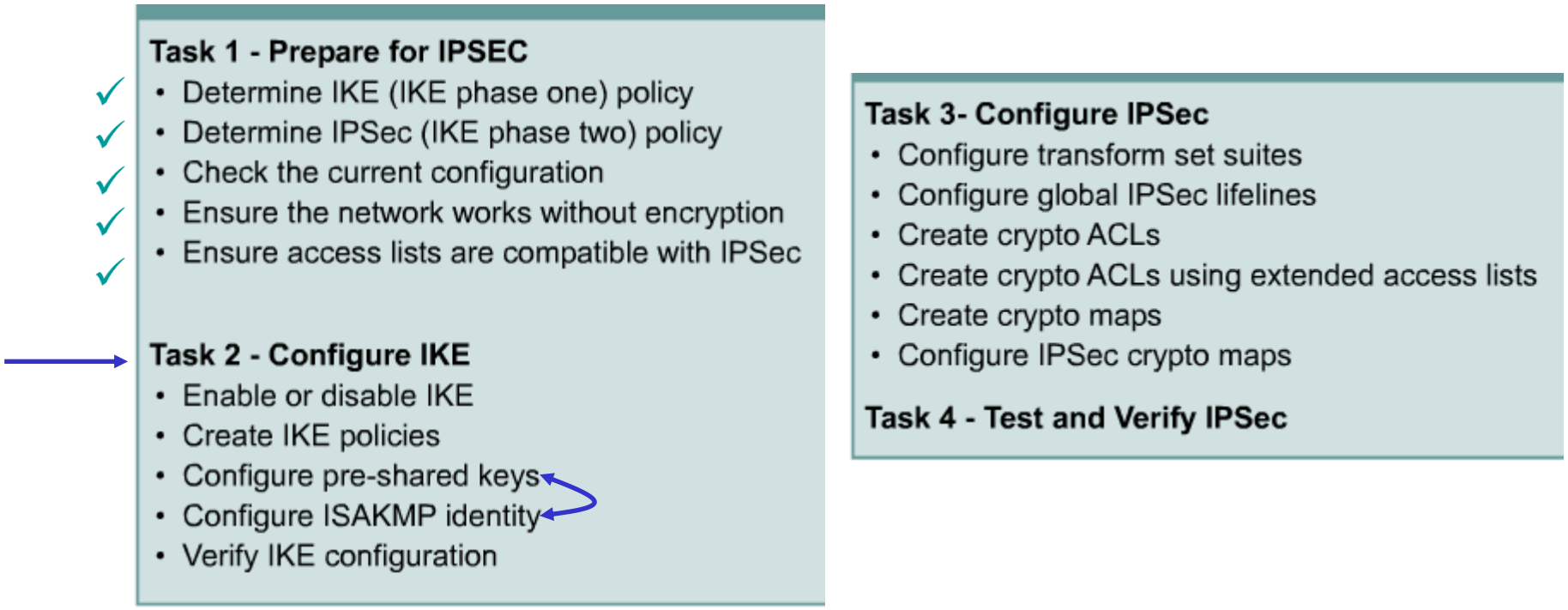
IPSec policy example



Policy	Host A	Host B
Transform set	ESP-DES, Tunnel	ESP-DES, Tunnel
Peer hostname	RouterB	RouterA
Peer IP address	172.30.2.2	172.30.1.2
Hosts to be encrypted	10.0.1.3	10.0.2.3
Traffic (packet) type to be encrypted	TCP	TCP
SA establishment	ipsec-isakmp	ipsec-isakmp

- The figure shows a summary of IPSec encryption policy details that will be configured in examples in this module.
- Details about IPSec transforms are covered in a later section in this module.
- The example policy specifies that TCP traffic between the hosts should be encrypted by IPSec using DES.

Configuring IKE



- By default, IKE is enabled in Cisco IOS.

Step 1 – Enable IKE

Task 2 - Configure IKE

- Enable or disable IKE
- Create IKE policies
- Configure pre-shared keys
- Configure ISAKMP identity
- Verify IKE configuration


```
router(config)#
[no] crypto isakmp enable
```

```
RouterA(config)#crypto isakmp enable
```

- This command globally enables or disables IKE at the router
- IKE is enabled by default
- IKE is enabled globally for all interfaces at the router
- Use the no form of the command to disable IKE
- An ACL can be used to block IKE on a particular interface

- By default, IKE is enabled in Cisco IOS.

Step 2 – Create IKE policies

Task 2 - Configure IKE
- Enable or disable IKE
- Create IKE policies
- Configure pre-shared keys
- Configure ISAKMP identity
- Verify IKE configuration


```
router(config)#
crypto isakmp policy priority
```

- Defines an IKE policy, which is a set of parameters used during IKE negotiation
- Invokes the config-isakmp command mode

```
RouterA(config)#crypto isakmp policy 110
```

```
RouterA(config-iskamp)#
```

Apply all options for IKE Phase One policy here.

- You then create your isakmp (Internet Security Association and Key Management Protocol) policies.
- The lower the policy-number or priority, the more preferred it is.
- You might have many policies on one device because each remote peer could have a different security profile created.
- For example…

Step 2 – Create IKE policies

Task 2 - Configure IKE
- Enable or disable IKE
- Create IKE policies
- Configure pre-shared keys
- Configure ISAKMP identity
- Verify IKE configuration


```
router(config)#
crypto isakmp policy priority
```

- Defines an IKE policy, which is a set of parameters used during IKE negotiation
- Invokes the config-isakmp command mode

```
RouterA(config)#crypto isakmp policy 110
```

```
RouterA(config-iskamp)#
```

Apply all options for IKE Phase One policy here.

- For example, **R&D lab** would probably have high security settings applied such as 3DES for payload encryption and SHA-1 for authentication and integrity.
- A **shipping office** might only have MD5 authentication and integrity and no encryption because it may not matter whether your competitors know how many items you are shipping.

Step 2 – Create IKE policies

Task 2 - Configure IKE
- Enable or disable IKE
- Create IKE policies
- Configure pre-shared keys
- Configure ISAKMP identity
- Verify IKE configuration


```
router(config)#
crypto isakmp policy priority
```

- Defines an IKE policy, which is a set of parameters used during IKE negotiation
- Invokes the config-isakmp command mode

```
RouterA(config)#crypto isakmp policy 110
```

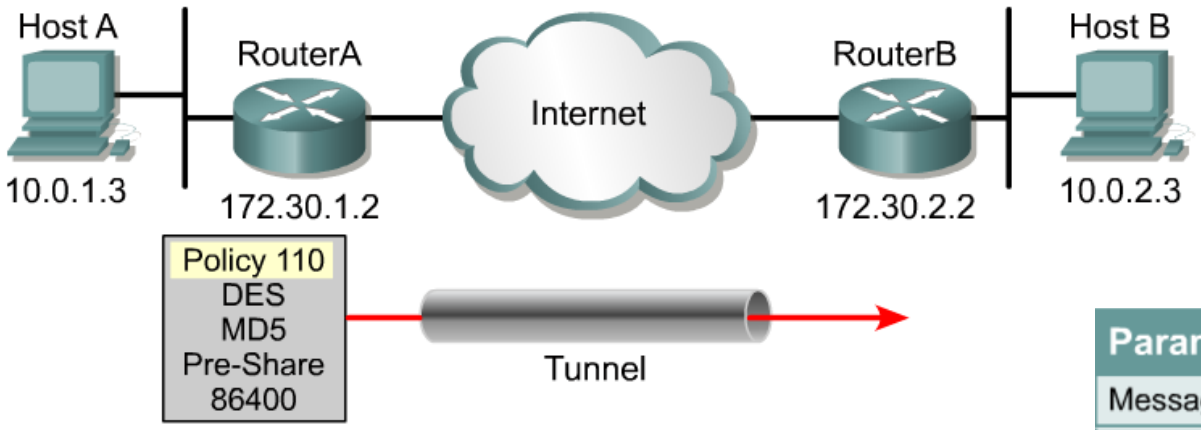
Parameter	Strong
Message encryption algorithm	DES
Message Integrity - hash algorithm	MD5
Peer authentication method	Pre-share
Key exchange parameters, Diffie-Hellman group identifier	D-H Group 1
ISAKMP-lifetime of established security association	86400 seconds

Policy	Host A	Host B
Transform set	ESP-DES, Tunnel	ESP-DES, Tunnel
Peer hostname	RouterB	RouterA
Peer IP address	172.30.2.2	172.30.1.2
Hosts to be encrypted	10.0.1.3	10.0.2.3
Traffic (packet) type to be encrypted	TCP	TCP
SA establishment	ipsec-isakmp	ipsec-isakmp

- The goal of defining a suite of IKE policies is to establish ISAKMP peering between two IPSec endpoints.
- Use the IKE policy details gathered during the planning task.

Create IKE policies with the crypto isakmp command

Task 2 - Configure IKE
- Enable or disable IKE
- Create IKE policies
- Configure pre-shared keys
- Configure ISAKMP identity
- Verify IKE configuration




```
router(config)#
crypto isakmp policy priority
```

- Defines the parameters within the IKE policy 110

```
RouterA(config)#crypto isakmp policy 110
RouterA(config-isakmp)#authentication pre-share
RouterA(config-isakmp)#encryption des
RouterA(config-isakmp)#group1
RouterA(config-isakmp)#hash md5
RouterA(config-isakmp)#lifetime 86400
```

Parameter	Strong
Message encryption algorithm	DES
Message Integrity - hash algorithm	MD5
Peer authentication method	Pre-share
Key exchange parameters, Diffie-Hellman group identifier	D-H Group 1
ISAKMP-lifetime of established security association	86400 seconds

- The **`crypto isakmp policy`** command invokes the ISAKMP policy configuration command mode (**`config-isakmp`**) where the ISAKMP parameters can be set.

Create IKE policies with the crypto isakmp command

Task 2 - Configure IKE
- Enable or disable IKE
- Create IKE policies
- Configure pre-shared keys
- Configure ISAKMP identity
- Verify IKE configuration

Keyword	Accepted Values	Default Values	Description
des	56 bit DES-CBS	des	Message encryption algorithm.
sha **md5**	SHA-1 (HMAC variant) MD5 (HMAC variant)	sha	Message integraity (Hash) algorithm.
rsa-sig **rsa-encr** **pre-share**	RSA signatuures RSA encrypted nonces pre-shared keys	rsa-sig	Peer authentication method.
1 **2**	768-bit Diffie-Hellman or 1024-bit Diffie-Helman	1	Key exchange parameters (Diffie-Hellman group identifier)
-	Can specify any number of seconds	86,400 seconds (one day)	ISAKMP-established SA's lifetime. You can usually leave this value at the default.
exit			Exits the config-isakmp mode

- If one of these commands is not specified for a policy, the default value will be used for that parameter.
- While in the **config-isakmp** command mode, the keywords are available to specify the parameters in the policy as shown.

IKE policy negotiation

Task 2 - Configure IKE
- Enable or disable IKE
- Create IKE policies
- Configure pre-shared keys
- Configure ISAKMP identity
- Verify IKE configuration


```
RouterA(config)#                    RouterB(config)#
•crypto isakmp policy 100           crypto isakmp policy 100
•  hash md5                           hash md5
•  authentication pre-share           authentication pre-share
•crypto isakmp policy 200           crypto isakmp policy 200
•  authentication rsa - sig           authentication rsa - sig
•  hash sha                           hash sha
•crypto isakmp policy 300           crypto isakmp policy 300
•  authentication pre -               authentication pre -
```

The first two policies in each router can be successfully negotiated while the last one cannot.

- A match is made when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values, and when the remote peer's policy specifies a lifetime less than or equal to the lifetime in the policy being compared.

Step 3 – Configure ISAKMP identity

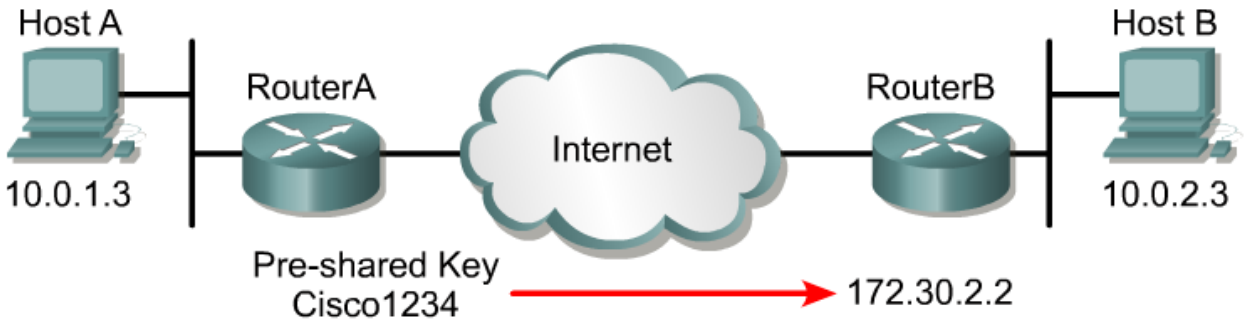

```
router(config)#
crypto isakmp identity {address | hostname}
```

- Defines whether ISAKMP identity is done by IP address or hostname
- Use consistency across ISAKMP peers

Command	Description
address	Sets the ISAKMP identity to the IP address of the interface that is used to communicate to the remote peer during ISAKMP negotiations. The keyword is typically used when there is only one interface that will be used by the peer for ISAKMP negotiations, and the IP address is known.
hostname	Sets the ISAKMP identity to the host name concatenated with the domain name (for example, **myhost.domain.com**). The keyword should be used if there is more than one interface on the peer that might be used for ISAKMP negotiations, or if the interface's IP address is unknown (such as with dynamically assigned IP addresses).

- IPSec peers authenticate each other during ISAKMP negotiations using the pre-shared key and the ISAKMP identity.
- The identity can either be the router IP address or host name.
- Cisco IOS software uses the IP address identity method by default.
- A command indicating the address mode does not appear in the router configuration.

Step 4 – Configure pre-shared keys

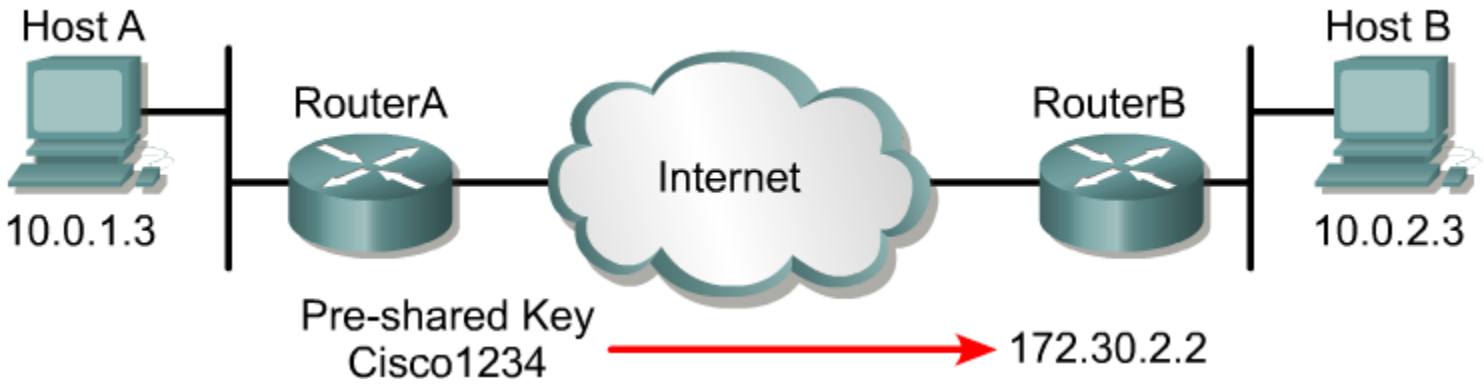



```
router(config)#
crypto isakmp key keystring address peer-address
router(config)#
crypto isakmp key keystring hostname hostname
RouterA(config)#crypto isakmp key cisco1234 address
 172.30.2.2
```

- Assigns a keystring and the peer address
- The peer's IP address or host name can be used

- Configure a pre-shared authentication key with the **`crypto isakmp key`** global configuration command.
- This key must be configured whenever pre-shared keys are specified in an ISAKMP policy.

Step 4 – Configure pre-shared keys



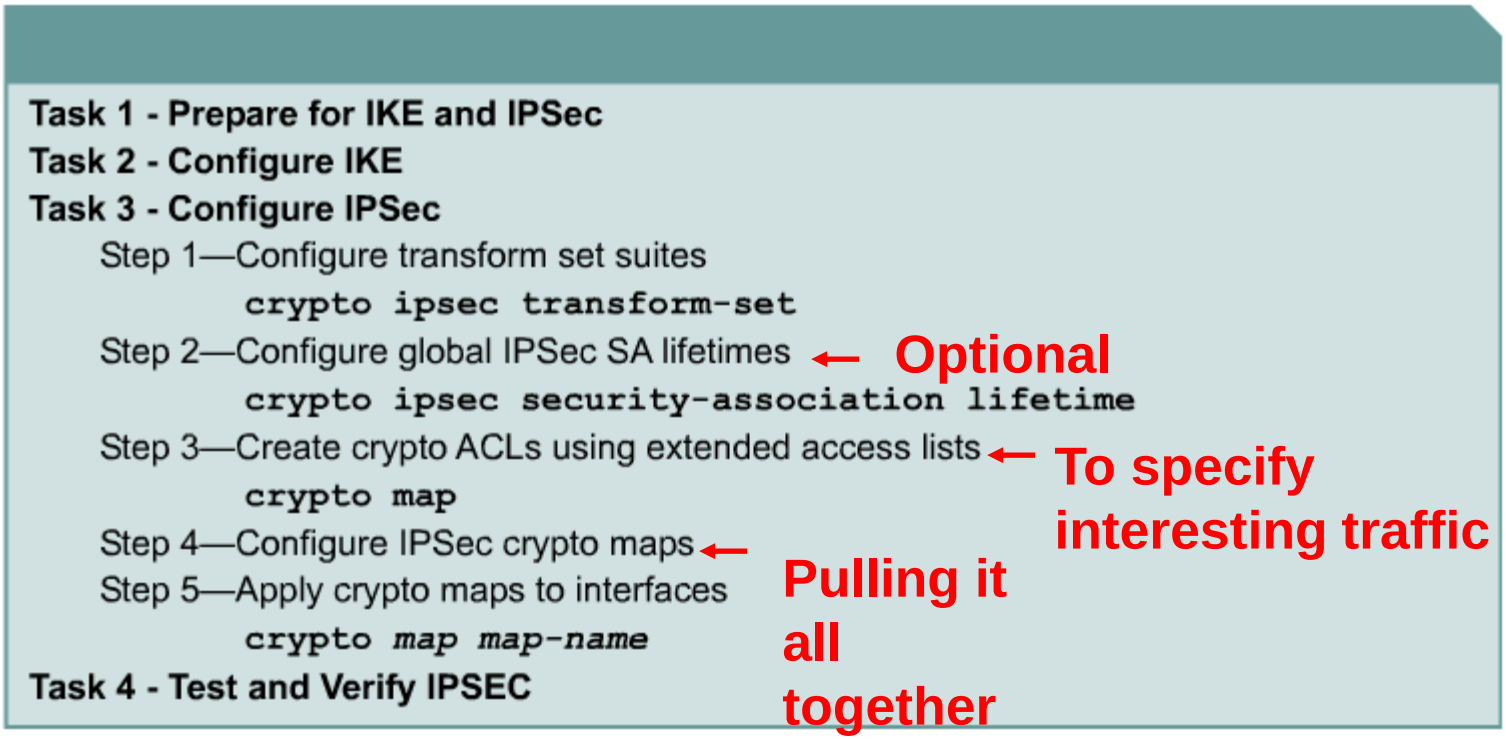

```
RouterA(config)#crypto isakmp key cisco1234 address 172.30.2.1
RouterA(config)#crypto isakmp policy 110
RouterA(config-isakmp)#hash md5
RouterA(config-isakmp)#authentication pre-share
```

```
RouterB(config)#crypto isakmp key cisco1234 address 172.30.1.1
RouterB(config)#crypto isakmp policy 110
RouterB(config-isakmp)#hash md5
RouterB(config-isakmp)#authentication pre-share
```

Step 5 – Verify IKE configuration

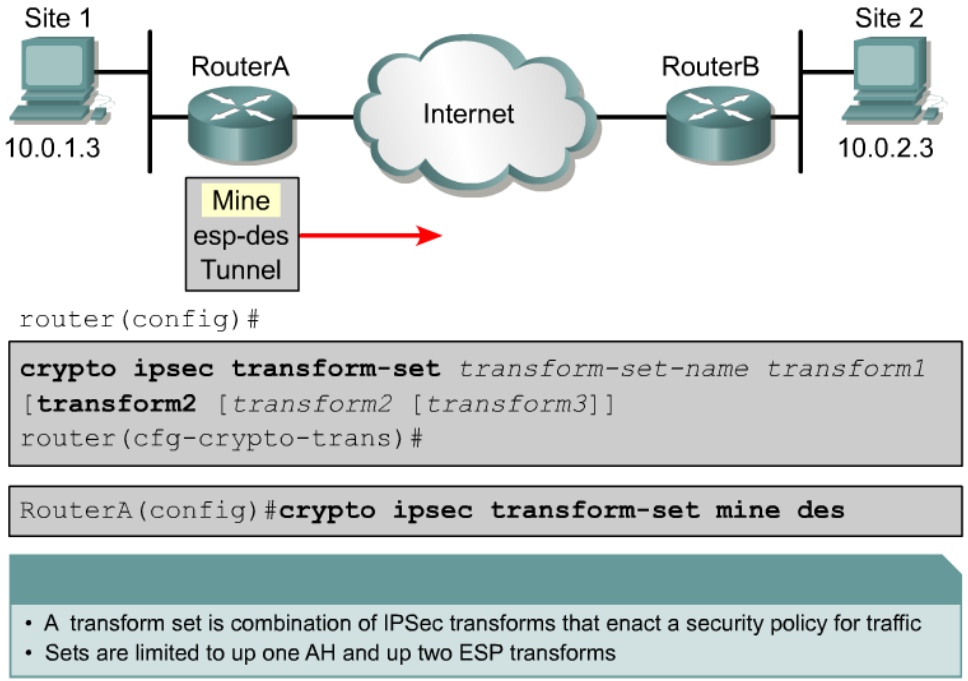

```
RouterA#show crypto isakmp policy
Protection suite of priority 110
encryption algorithm: DES - Data Encryption Standard (56 bit keys).
hash algorithm: Message Digest 5
authentication method: Pre-Shared Key
Diffie-Hellman group: #1 (768 bit)
lifetime: 86400 seconds, no volume limit
Default protection suite
encryption algorithm: DES - Data Encryption Standard (56 bit keys).
hash algorithm: Secure Hash Standard
authentication method: Rivest-Shamir-Adleman Signature
Diffie-Hellman group: #1 (768 bit)
lifetime: 86400 seconds, no volume limit
```

Task 3 – Configure IPSec



- The general tasks and commands used to configure IPSec encryption on Cisco routers are summarized as follows.

Step 1 – Configure transform set suites



- A **transform set** defines the type of authentication, integrity, and payload encryption you will use for your VPN tunnel.
- Depending upon you r security policy, you can choose what type of algorithms will be applied to the data for a specific connection.
- You could choose just authentication and integrity by selecting AH, or you could choose payload encryption, authentication, and integrity by selecting two ESP options, or you could choose to have both AH and ESP options.

Step 1 – Configure transform set suites

- When you create transform sets, you have the options that are outlined here:

Transform Type	AH Transform	ESP Encryption Transform	ESP Authentication Transform
Allowed Transform Options	**ah-md5-hmac** **ah-sha-hmac**	**esp-des** **esp-3des** **esp-null**	**esp-md5-hmac** **esp-sha-hmac**

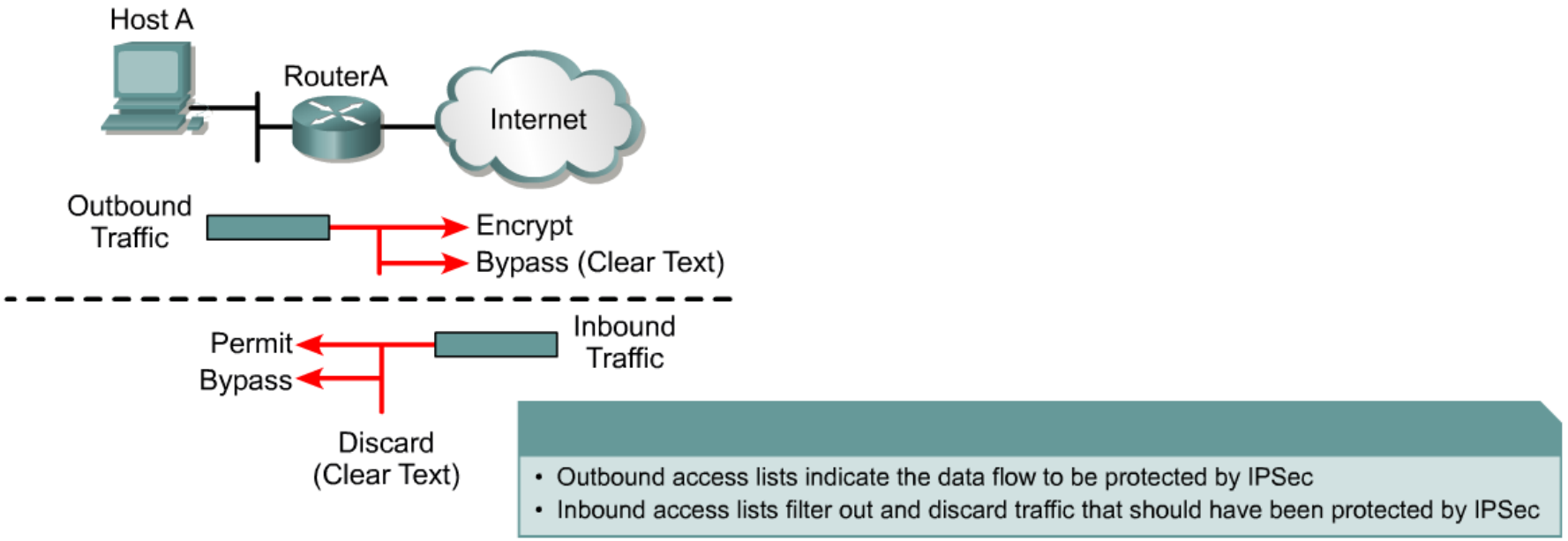
Step 2 – Configure global IPSec security association lifetimes (optional)


```
router(config)#
crypto ipsec security-association lifetime
     {seconds seconds | kilobytes kilobytes}
```

```
RouterA(config)#crypto ipsec security-association
lifetime 86400
```

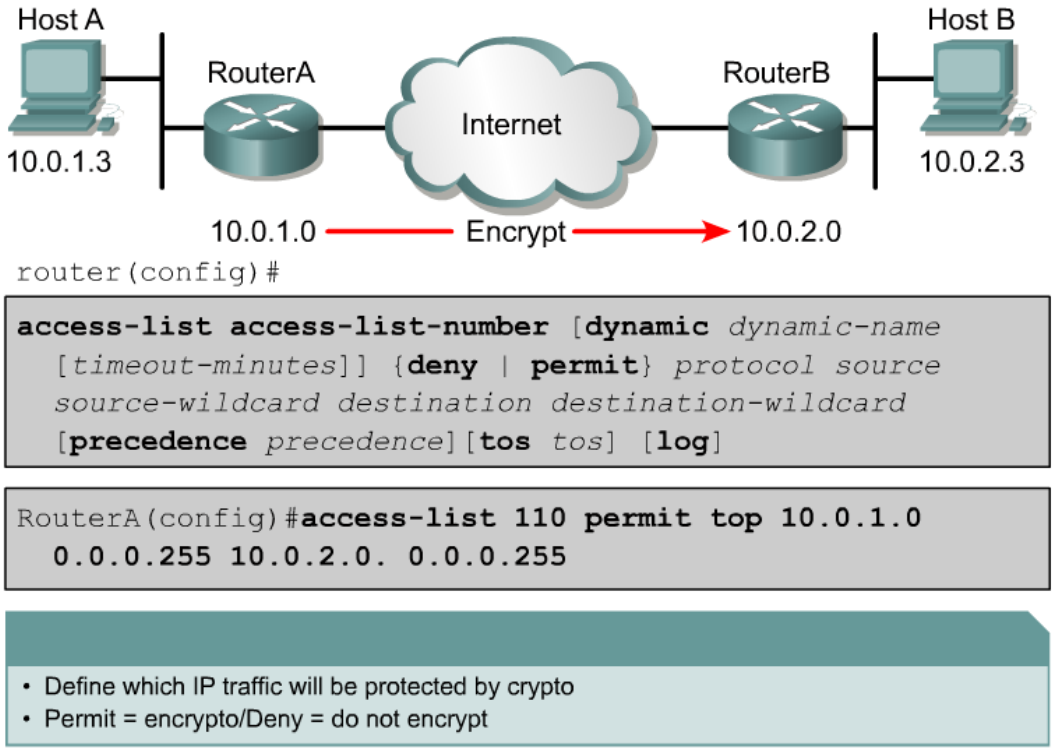
- Configures global IPSec SA lifetime values used when negotiating IPSec security associations
- IPSec SA lifetimes are negotiated during IKE phase 2
- Can optionally configure interface specific IPSec SA lifestyles in crypto maps
- IPSec SA lifetimes in crypto maps override global IPSec SA lifetimes

Step 3 - Crypto ACLs



- So far, you have configured how all of the traffic will be encrypted and transported, and what your peers will use as session keys.
- All that is left is to specify “interesting traffic” to be encrypted by your tunnel.
- This is done using extended ACLs.

Step 3 – Create crypto ACLs using extended access lists



- The crypto ACL does not “permit” or “deny” traffic as normal ACLs do.
- It is used to dfine what is encrypted “permitted” or not encrypted “denied” in your VPN tunnel.
- All traffic still flows from from device to device unless a normal ACL has been used to do otherwise.

Step 3 – Create crypto ACLs using extended access lists


```
router(config)#
access-list access-list-number [dynamic dynamic-name
  [timeout-minutes]] {deny | permit} protocol source
  source-wildcard destination destination-wildcard
  [precedence precedence][tos tos] [log]
```

```
RouterA(config)#access-list 110 permit top 10.0.1.0
  0.0.0.255 10.0.2.0. 0.0.0.255
```

- Define which IP traffic will be protected by crypto
- Permit = encrypto/Deny = do not encrypt

- The crypto ACL must mirror images of each other..

Step 3 – Create crypto ACLs using extended access lists

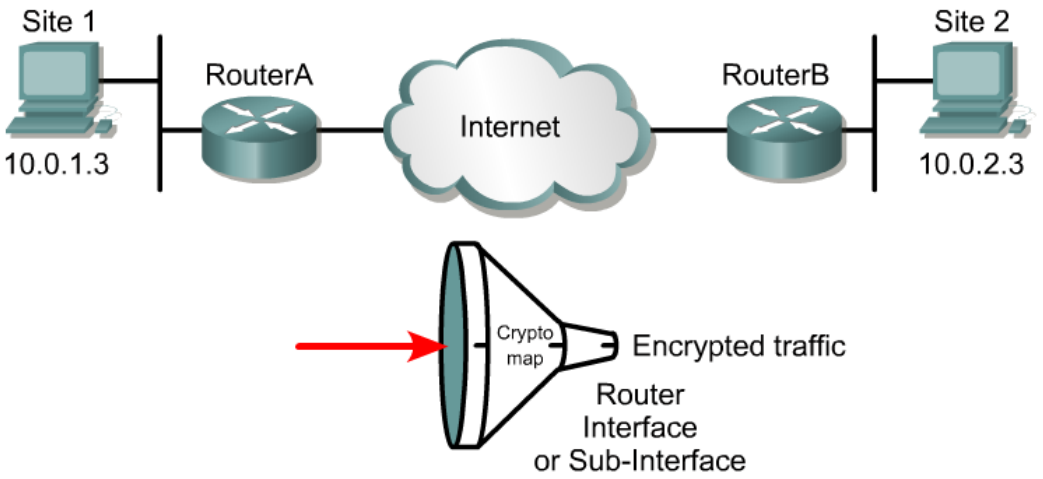

```
RouterA(config)#access-list 110 permit tcp
    10.0.1.0 0.0.0.255  10.0.2.0 0.0.0.255

RouterB(config)#access-list 101 permit tcp
    10.0.2.0 0.0.0.255  10.0.1.0 0.0.0.255
```

Mirror image ACLs must be configured

- The crypto ACL must mirror images of each other.

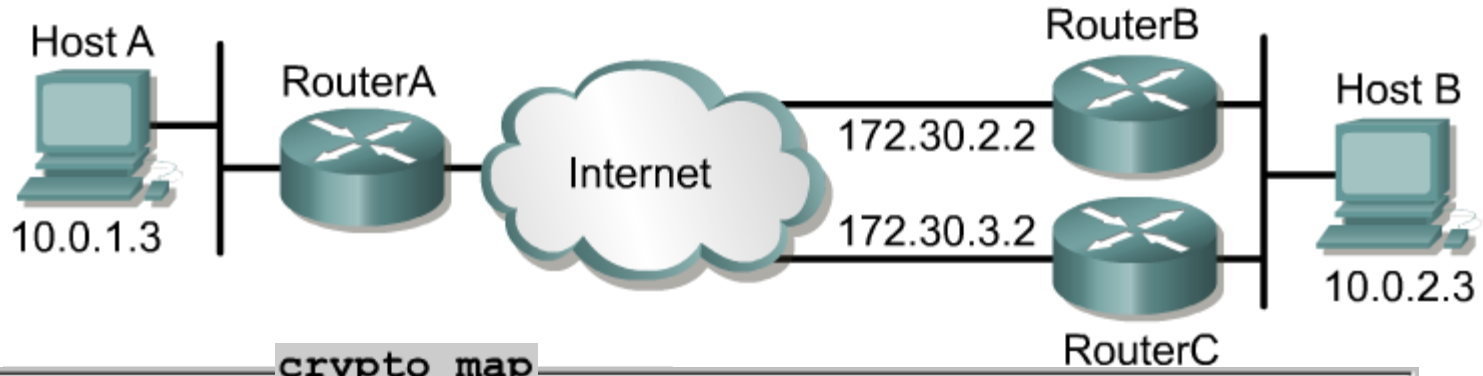
Step 4 - Purpose of crypto maps



Crypto maps define the following:
- The access list to be used
- Remote VPN peers
- Transform set to be used
- Key management method
- Security association lifetimes

- Now that you have all the required information to create your VPN tunnel, you need to pull everything together and apply it to an interface.
- Crypto maps do this.

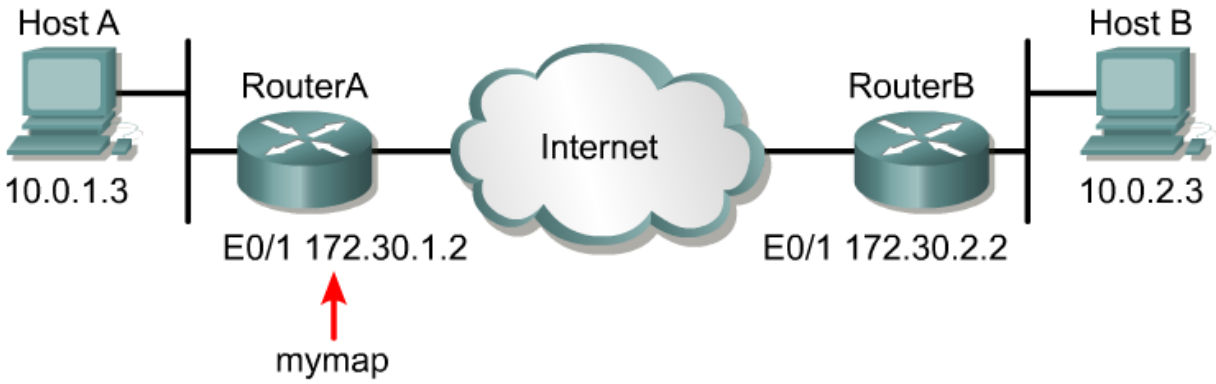
Example crypto map commands




```
RouterA(config)#crypto map symap 110 ipsec-isakmp
RouterA(config-crpyto-map)#match address 110
RouterA(config-crpyto-map)#set peer 172.30.2.2
RouterA(config-crpyto-map)#set peer 172.30.3.2
RouterA(config-crpyto-map)#set pfs group1
RouterA(config-crpyto-map)#set transform-set mine
RouterA(config-crpyto-map)#set security-assoctiona
lifetime 86400
```

- There are a few types of crypto map statements, but for this exam the crypto map ipsec-isakmp is used for automatic key exchanges.

Step 5 – Apply crypto maps to interfaces




```
router(config-if)#
crypto map map-name
```

```
RouterA(config)#interface fastethernet0/1
RouterA(config-if)#crypto map mymap
```

- Apply the crypto map to outgoing interface
- Activates the IPSec policy

- Once you have specified all of the needed information, you need to apply it to the outgoing interface.

IPSec configuration examples


```
RouterA#show running config
crypto ipsec transform-set mine
esp-des
!
crypto map mymap 10 ipsec-isakmp
set peer 172.30.2.2
set transform-set mine
match address 110
!
interface Ethernet 0/1
ip address 172.30.1.2 255.255.255.0
no ip directed-broadcast
crypto map mymap
!
access-list 110 permit tcp 10.0.1.0
0.0.0.255 10.0.2.0 0.0.0.255
```

```
RouterB#show running config
crypto ipsec transform-set mine
esp-des
!
crypto map mymap 10 ipsec-isakmp
set peer 172.30.1.2
set transform-set mine
match address 110
!
interface Ethernet 0/1
ip address 172.30.2.2 255.255.255.0
no ip directed-broadcast
crypto map mymap
!
access-list 110 permit tcp 10.0.2.0
0.0.0.255 10.0.1.0 0.0.0.255
```

Test and Verify IPSec

Task 1 - Prepare for IKE and IPSEC
Task 2 - Configure IKE
Task 3 - Configure IPSec
Task 4 - Test and Verify IPSEC

- Display configured IKE poliicies.
  `show crypto isakmp policy` (`show iskmlp policy` on a PIX)
- Display configured transform sets.
  `show crypto ispec transform set`
- Display Phase | security associations
  `show crypto isakmp sa` (`show isakmp sa` on a PIX)
- Display the current state of your IPSec SAs.
  `show crypto ipsec sa`
- Display configured crypto maps.
  `show crypto map`
- Enable debug output for IPSec events.
  `debug crypto ipsec`
- Enable debug output for ISAKMP events.
  `debug crypto isakmp`