

# Virtual Private Networks - VPNs

9-dec-2014

# What this lecture is about:

---

- ▶ Purpose and types of VPNs.
- ▶ IPsec
  - ▶ Algorithms behind IPsec
  - ▶ Configuring an IPsec site-to-site VPN
- ▶ Remote access and software VPN clients.

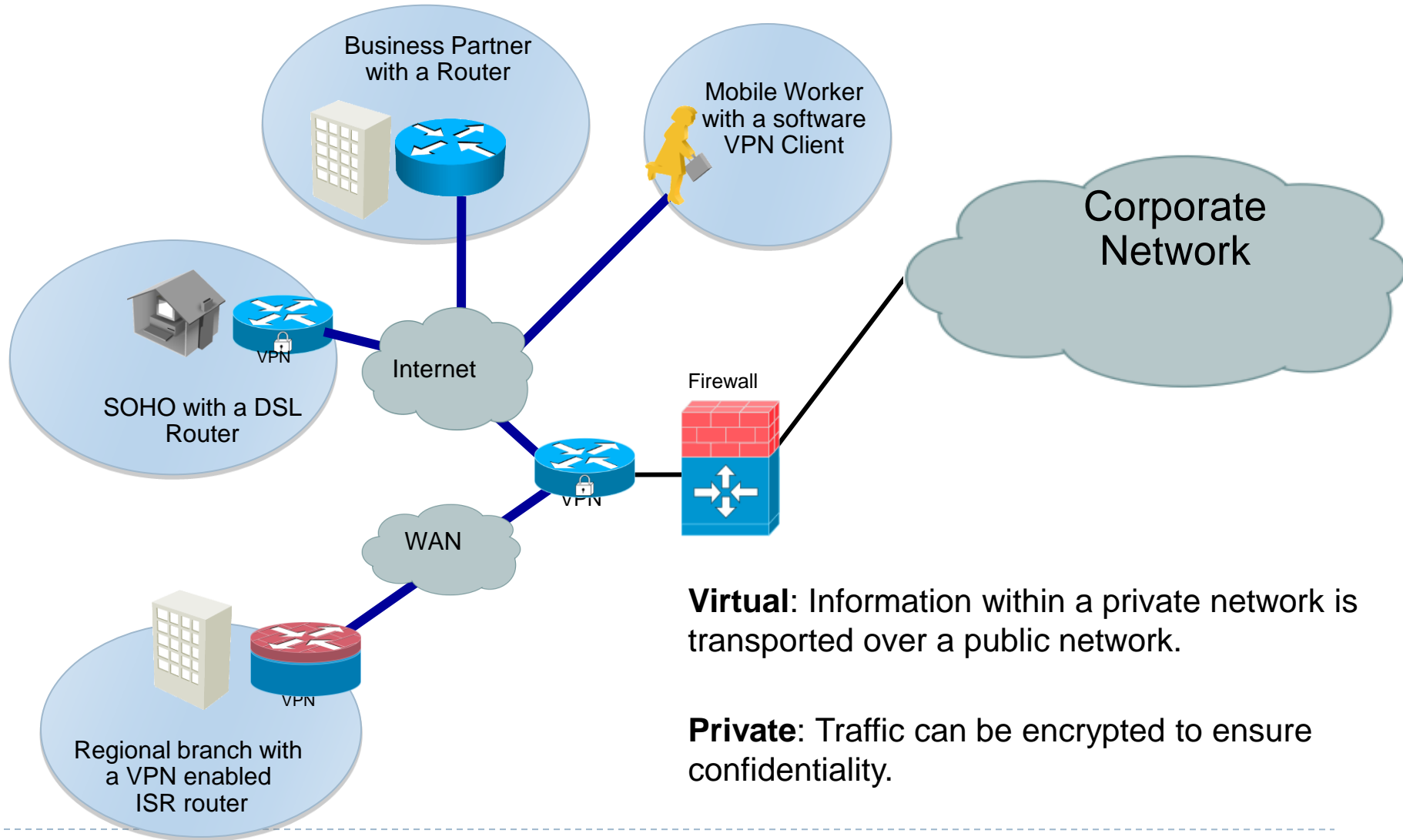


# What is a VPN?

---

- ▶ A VPN is an end-to-end private network connection over a third-party public network, such as the Internet.
- ▶ A VPN does not guarantee confidentiality by itself.
  - ▶ Cryptographic methods are used
  - ▶ A VPN becomes a tunnel carrying encrypted information
  - ▶ Can also ensure data authenticity
- ▶ IPsec is a security framework for VPNs
  - ▶ Defines several protocols that ensure privacy, integrity and authenticity.

# Where can we implement VPNs?



**Virtual:** Information within a private network is transported over a public network.

**Private:** Traffic can be encrypted to ensure confidentiality.

# VPN benefits

---

- ▶ THE main advantage of VPNs:
  - ▶ Providing a secure and isolated connection without requiring a dedicated physical connection.
- ▶ **Security:** VPNs use advanced encryption and authentication protocols.
- ▶ **Scalability:** VPNs use the existing infrastructure of the Internet.
  - ▶ Adding new users and networks is easy.
- ▶ **Compatibility:** VPNs can traverse any number of different connections.
  - ▶ LANs, WLANs, WANs, GSM networks, etc.
- ▶ **Cost-effectiveness:** VPNs do not require dedicated links and can work even without specialized hardware.

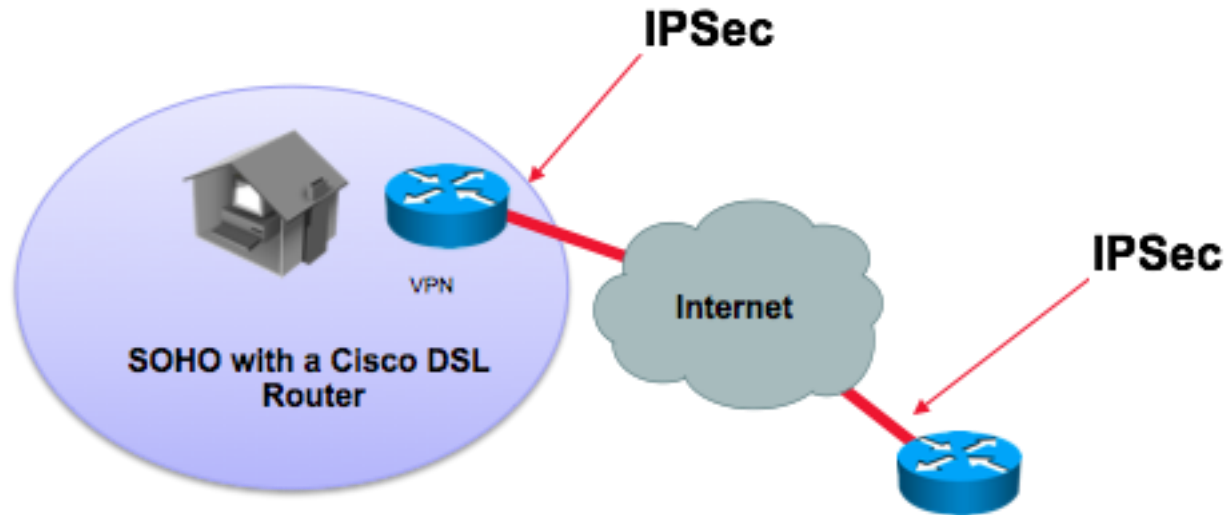
# Layer 3 VPN

---

- ▶ A VPN is a tunnel that connects two endpoints over a public network.
  - ▶ In order to get data to destination, a header must be added to all packets traveling through the tunnel.
  - ▶ This header provides all the benefits of a VPN.
  - ▶ This header also contains addressing information that allows all packets to reach destination.
- ▶ VPNs can be implemented at Layers 2, 3 and 5.
- ▶ We will be discussing about Layer 3 VPNs.
  - ▶ So the tunnel header will have some Layer 3 information.

# Layer 3 VPN

---



- ▶ Types of L3 VPN: GRE, IPsec, MPLS.
- ▶ The protection of data in a VPN is provided by the IPsec security framework.
- ▶ Encryption devices and VPN-aware services must be deployed on both ends of a VPN.
  - ▶ Intermediary devices are not even aware of the type of traffic they are carrying.

# VPN topologies

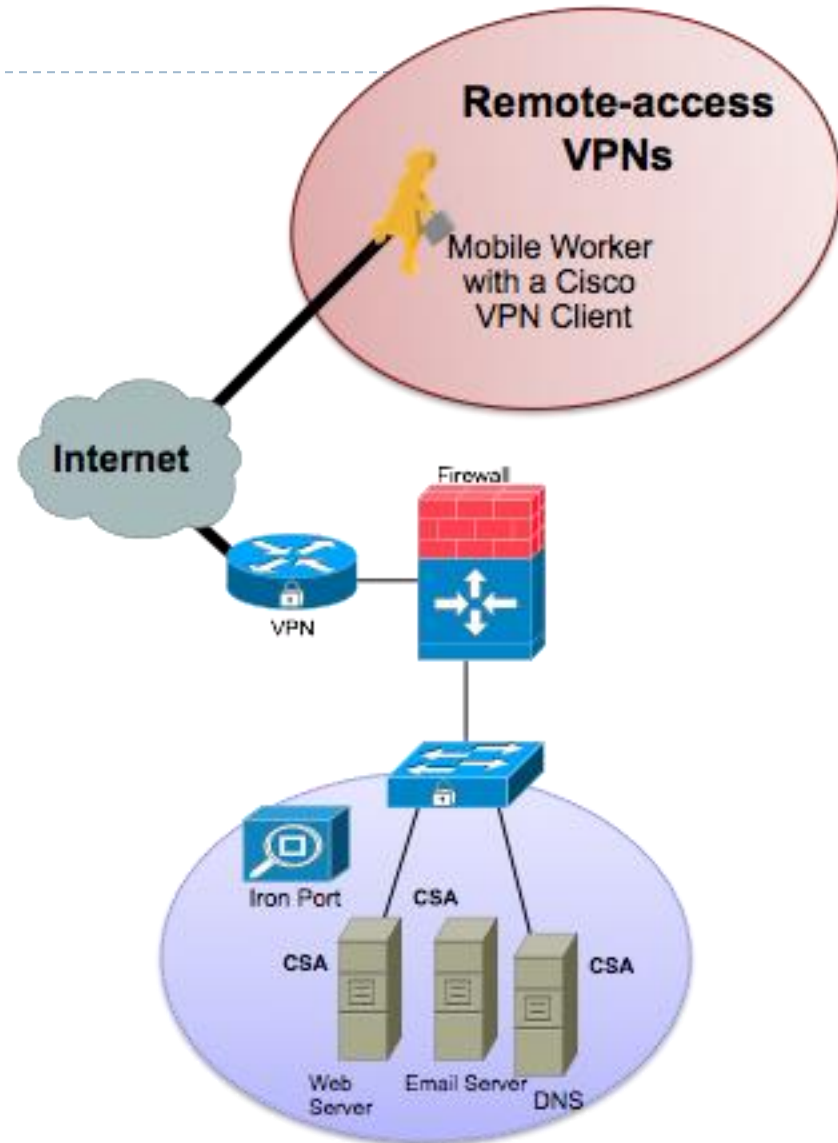
---

- ▶ There are two types of different VPN topologies:
- ▶ Remote-access VPNs
  - ▶ Remote users must have a broadband Internet connection
  - ▶ The VPN parameters are dynamically negotiated
  - ▶ The user establishes a VPN tunnel through the ISP
  - ▶ The tunnel is established only when required
  - ▶ Costs are associated only with the Internet connection's cost
- ▶ Site-to-site VPNs
  - ▶ Configured between two VPN-aware devices on both ends
  - ▶ Always-on
  - ▶ Provides interconnectivity between multiple networks on both sites.
  - ▶ Each end of the tunnel acts as a gateway for its networks.

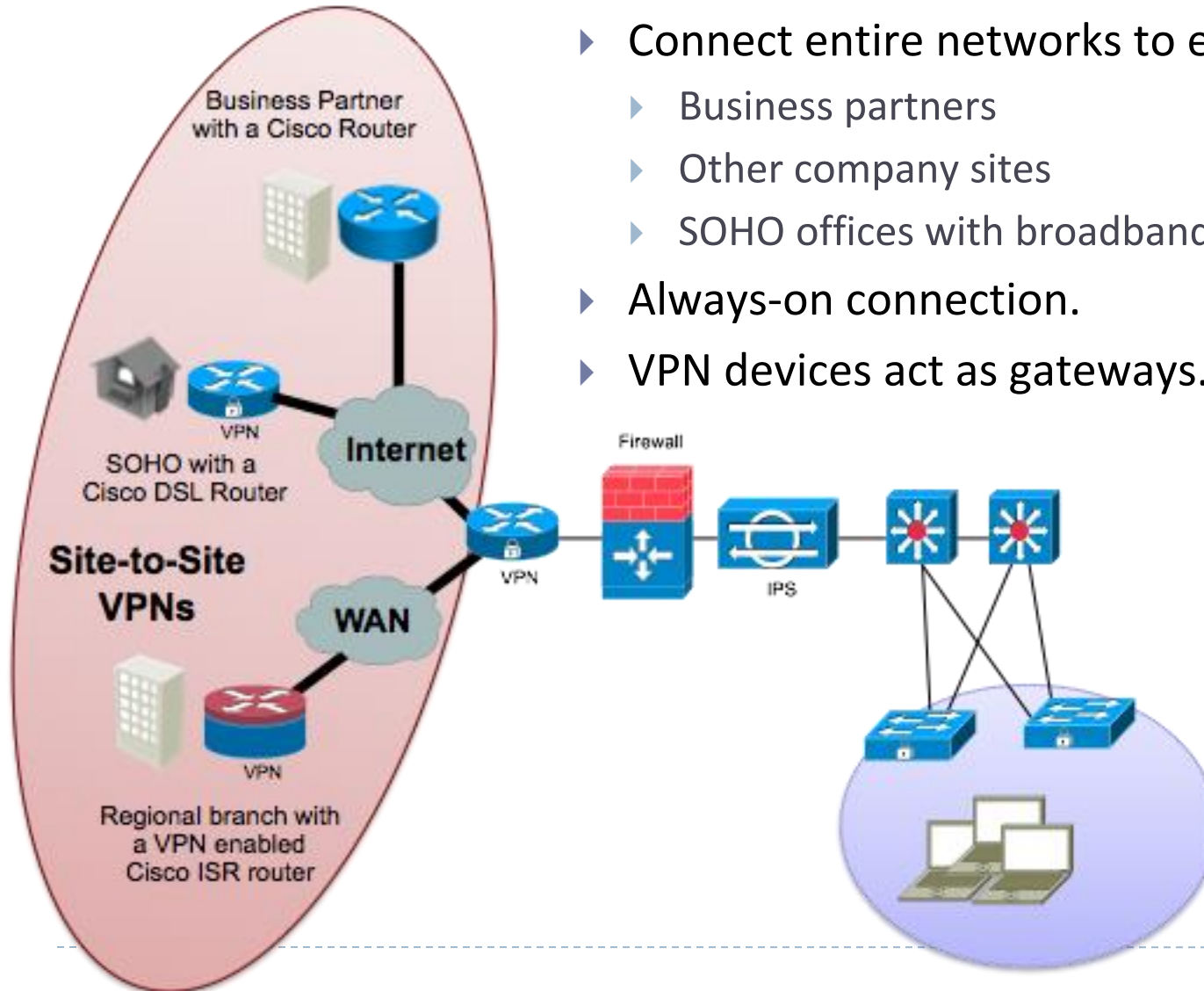


# Remote-access VPN

- ▶ Replacement for old dial-up or ISDN connections.
  - ▶ Past alternatives for dedicated secure connections to a central site.
- ▶ Requires a VPN client software
  - ▶ Like Cisco VPN Client
- ▶ Feasible for a single host
  - ▶ Or a small network



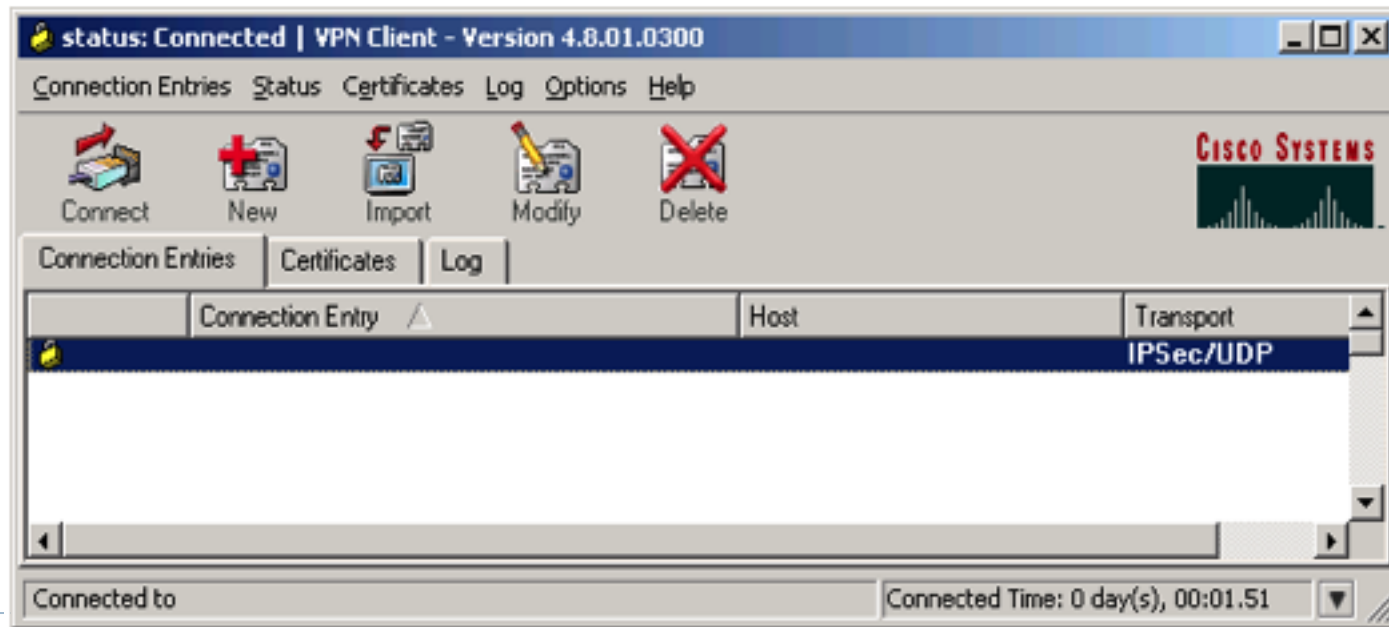
# Site-to-site VPN



- ▶ Connect entire networks to each other.
  - ▶ Business partners
  - ▶ Other company sites
  - ▶ SOHO offices with broadband connections
- ▶ Always-on connection.
- ▶ VPN devices act as gateways.

# VPN client software

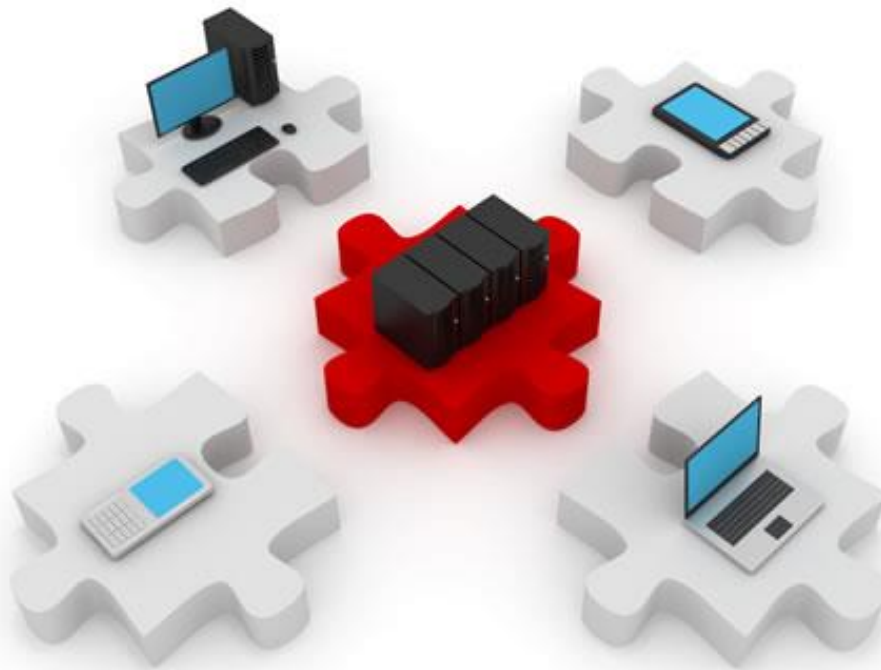
- ▶ In a remote-access VPN, each host has a VPN client software.
- ▶ All tunnel processing on the client side is done in software.
- ▶ The tunnel can become the host's new default gateway.
  - ▶ Or the host can be configured to encrypt only certain types of traffic (not everything will be sent through the tunnel).



# SSL VPN

---

- ▶ Emerging remote-access technology
- ▶ Only an SSL-capable browser is required.
  - ▶ The VPN is established using only the native SSL capabilities of a browser.
  - ▶ Provides access to TCP applications without a VPN software.
  - ▶ All processing is done in software.
  - ▶ Easiest to implement.
- ▶ Two modes of access:
  - ▶ clientless (described above)
  - ▶ thin client
    - ▶ The user is required to download a small Java applet.



# GRE Tunnels

# GRE encapsulation

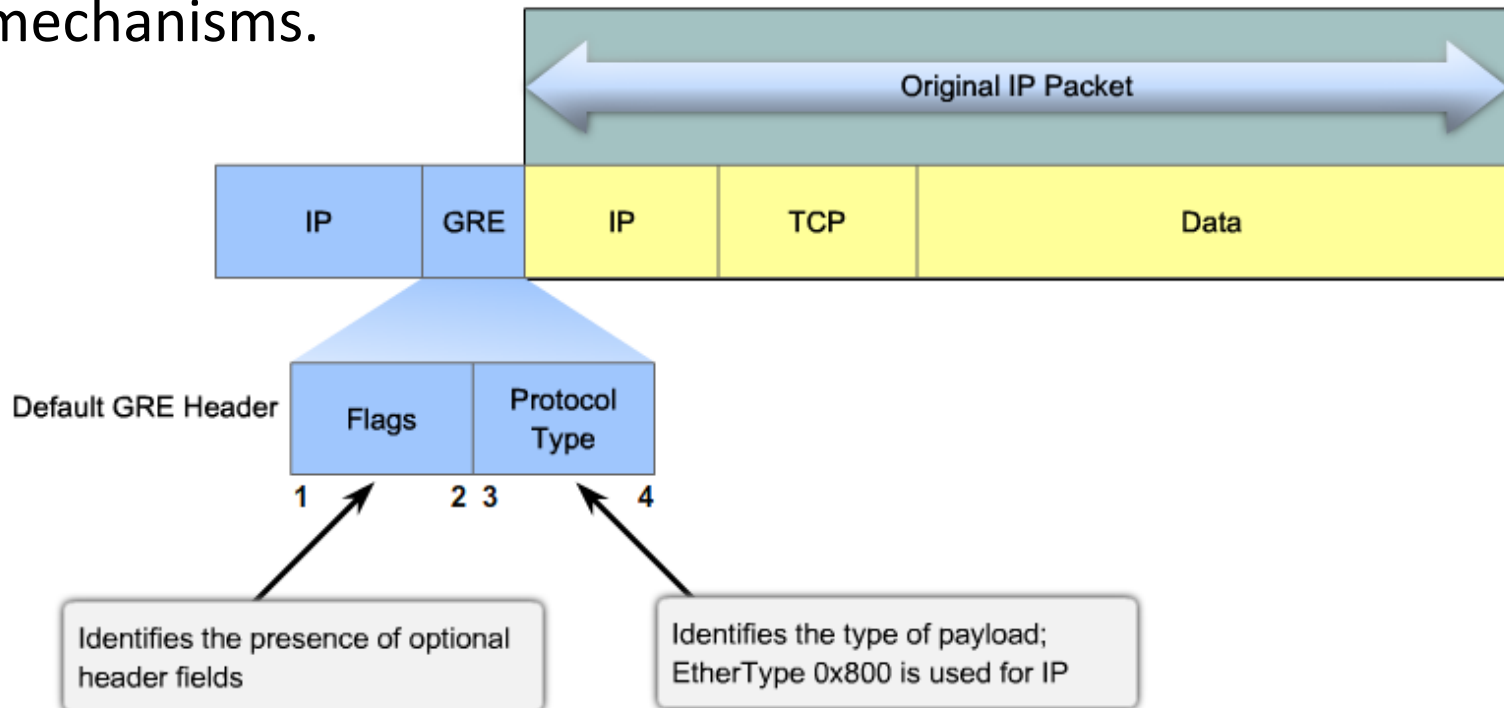
---

- ▶ **GRE (Generic Routing Encapsulation): RFC 1702 and 2784**
  - ▶ Is an OSI layer 3 tunneling protocol
  - ▶ Originally developed by Cisco, now a worldwide standard.
- ▶ **Can encapsulate multiple protocol packet types inside an IP tunnel.**
  - ▶ Adds an additional header between the tunnel's layer 3 header and the payload.
  - ▶ This header identifies the encapsulated protocol.

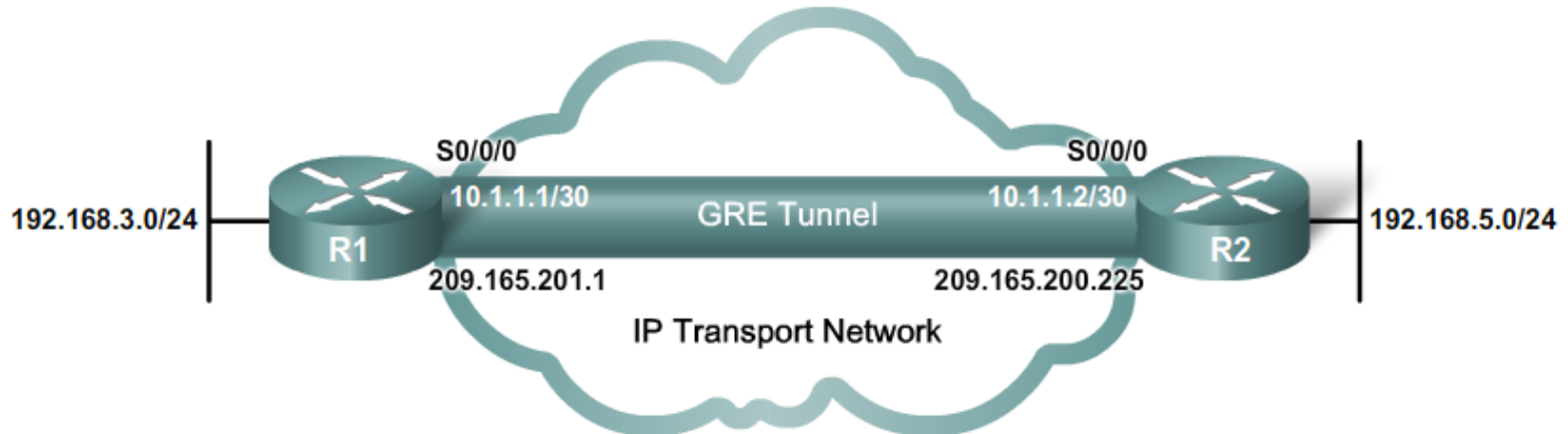


# GRE tunnel header

- ▶ GRE tunnels are stateless
  - ▶ Endpoints do not maintain information about the state or the availability of each other.
- ▶ They do not provide strong authentication and confidentiality mechanisms.



# GRE tunnel configuration



```
R1(config)# interface tunnel 0
R1(config-if)# ip address 10.1.1.1 255.255.255.252
R1(config-if)# tunnel source serial 0/0/0
R1(config-if)# tunnel destination 209.165.200.225
R1(config-if)# tunnel mode gre ip
R1(config-if)#
```

```
R2(config)# interface tunnel 0
R2(config-if)# ip address 10.1.1.2 255.255.255.252
R2(config-if)# tunnel source serial 0/0/0
R2(config-if)# tunnel destination 209.165.201.1
R2(config-if)# tunnel mode gre ip
R2(config-if)#
```

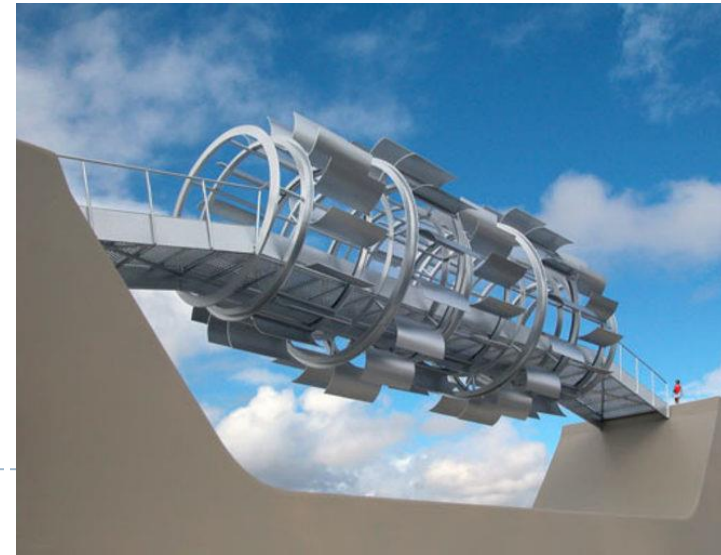
- ▶ The tunnel endpoints are virtual interfaces.
- ▶ The tunnel maps to a physical local interface and connects to a remote interface.
- ▶ The tunnel is a separate subnet by itself.
- ▶ Specifying the tunnel mode is optional - **GRE is the default mode** for any tunnel.



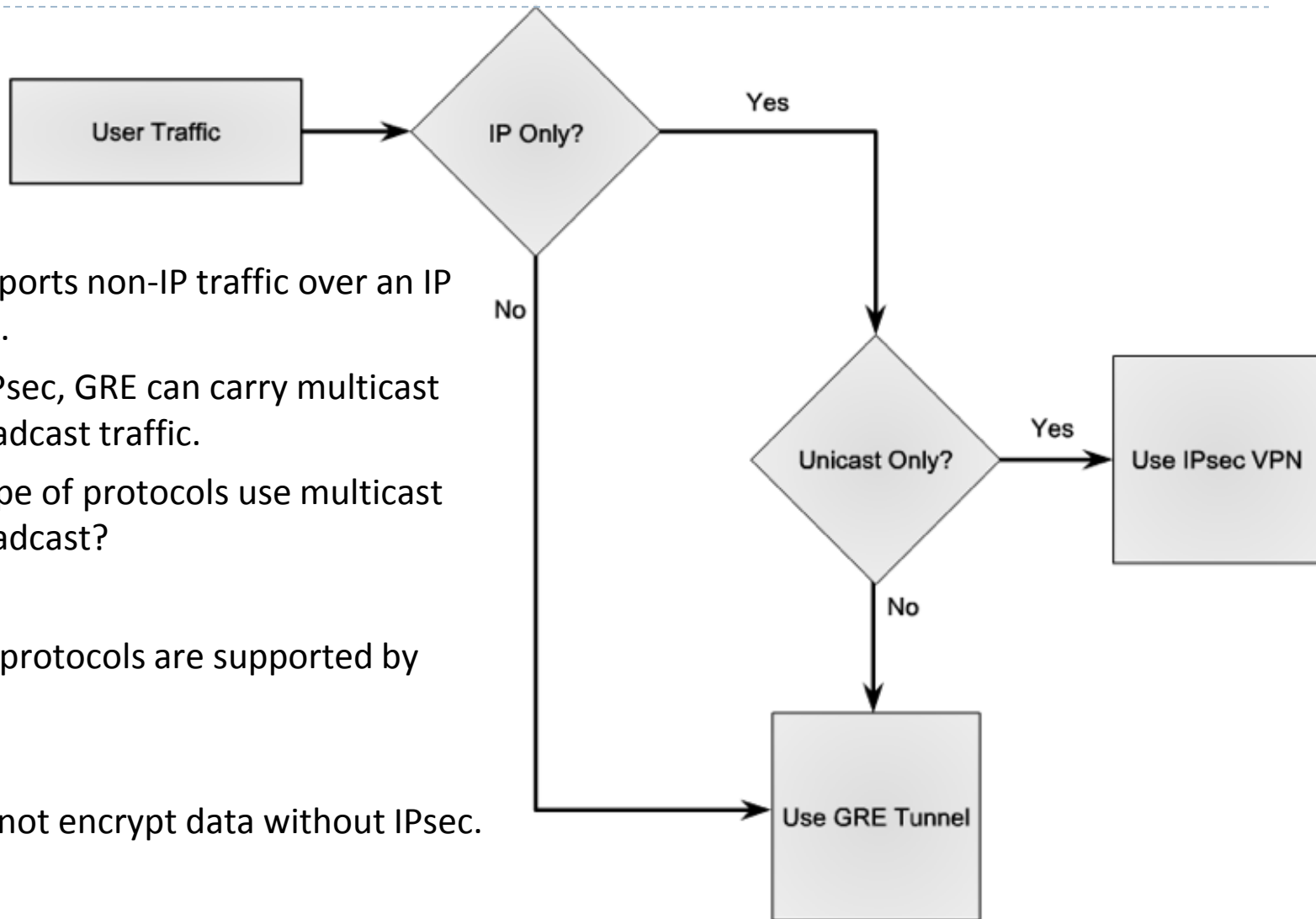
# GRE tunnels troubleshooting

---

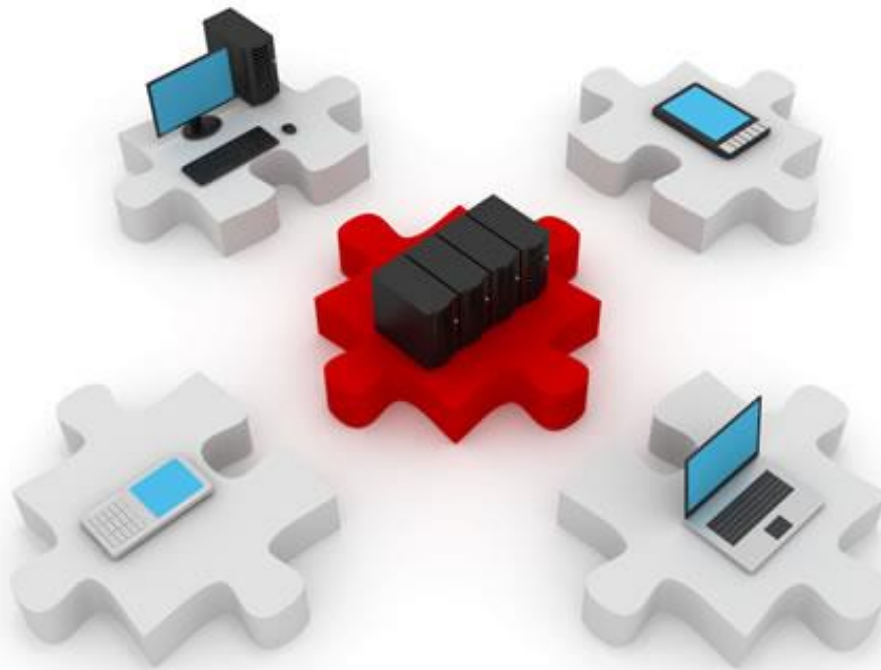
- ▶ A GRE tunnel might not become operational for a different number of reasons. Check the following:
  - ▶ The tunnel destination must be a reachable IP address (it must be present in the routing table).
  - ▶ The tunnel must have a valid source and a valid destination.
  - ▶ GRE tunnel traffic might be blocked by an ACL or firewall implementation.
  - ▶ The tunnel mode must be the same at both ends.



# GRE traffic support



- ▶ GRE supports non-IP traffic over an IP network.
- ▶ Unlike IPsec, GRE can carry multicast and broadcast traffic.
- ▶ What type of protocols use multicast and broadcast?
- ▶ Routing protocols are supported by GRE.
- ▶ GRE cannot encrypt data without IPsec.



# The IPsec Framework

# What is IPsec?

---

- ▶ IPsec is an IETF standard (RFC 2401-2412)
  - ▶ Defines ways to deploy VPNs using the IP addressing protocol.
  - ▶ Is a framework of open standards that describe how to secure communication.
- ▶ Relies on existing algorithms to provide:
  - ▶ Encryption
  - ▶ Authentication
  - ▶ Data integrity
  - ▶ Secure key exchange
- ▶ Can work over any L2 connection.



# IPsec topology

---



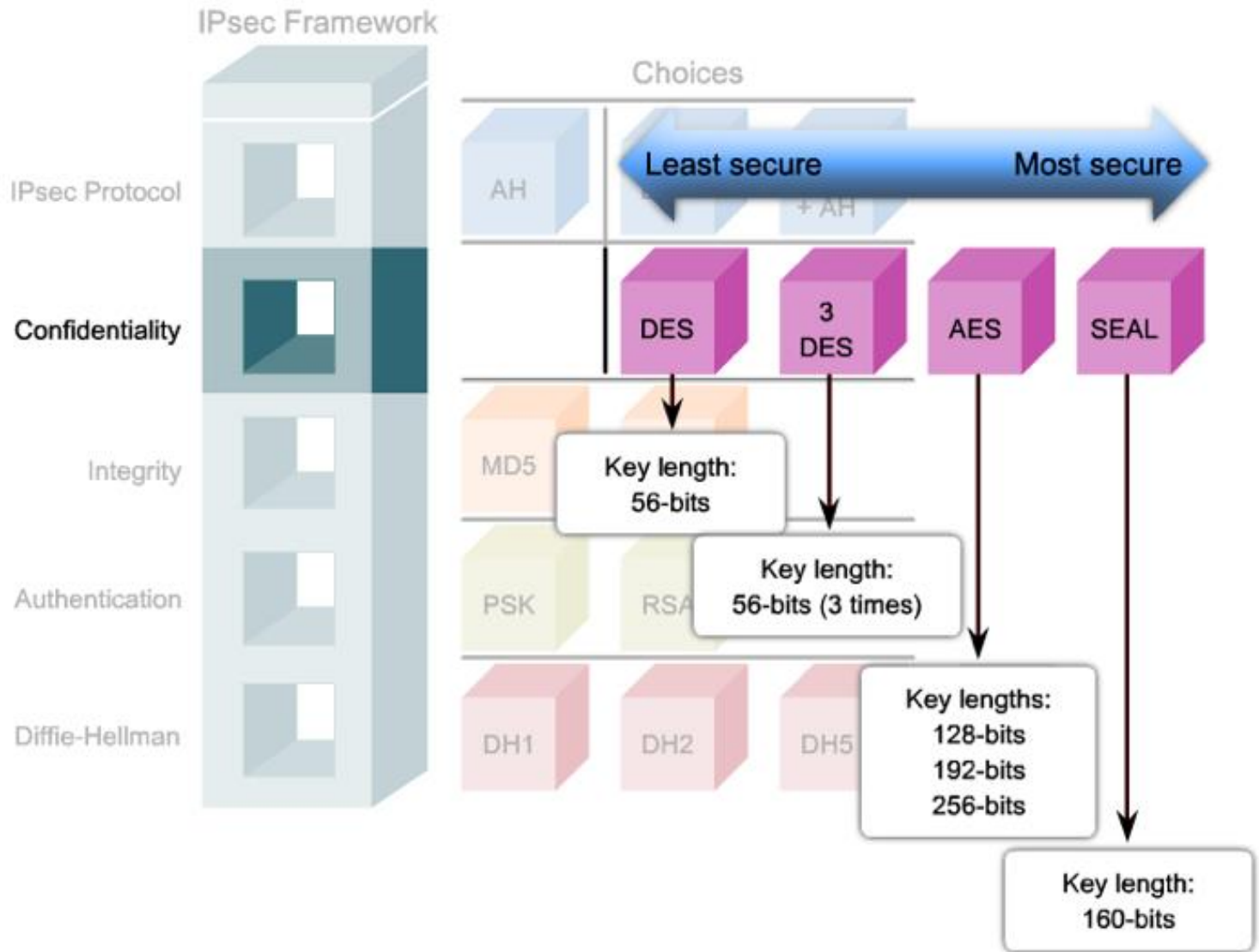
- ▶ IPsec works at the network layer, protecting and authenticating IP packets.
  - ▶ **IPsec only provides the framework**, the administrator chooses which algorithms will be used, depending on security requirements.
-

# IPsec building blocks

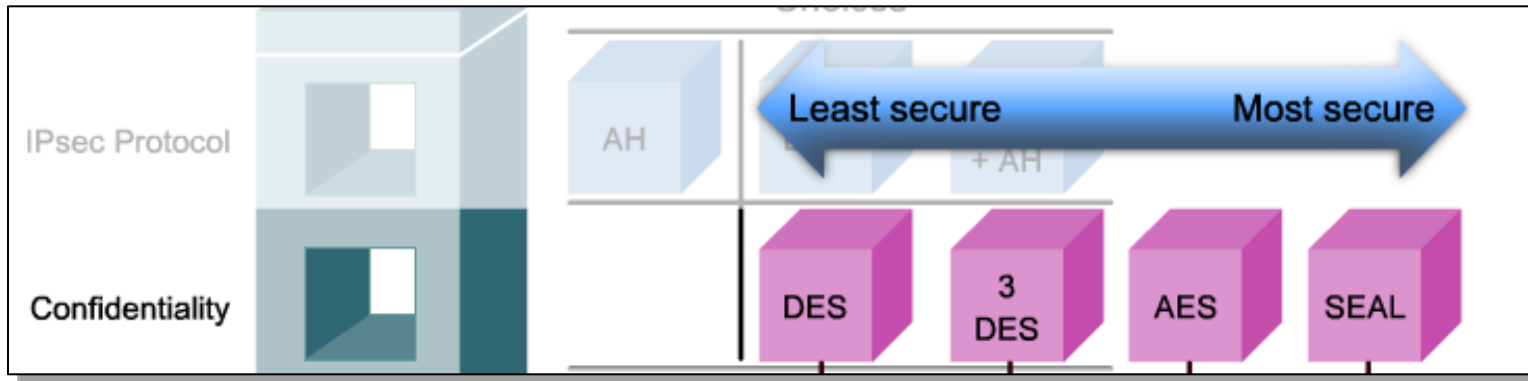
---

- ▶ IPsec protocol in use
  - ▶ Choices are: AH (Authentication Header) and ESP (Encapsulating Security Payload)
- ▶ Algorithms that provide confidentiality (encryption):
  - ▶ Examples: DES, 3DES, AES, SEAL
- ▶ Algorithms that ensure integrity:
  - ▶ Examples: MD5, SHA, along with other versions
- ▶ Algorithms that define the authentication method:
  - ▶ Choices include: pre-shared keys (PSK) or digitally signed using RSA.
- ▶ The mechanism to securely communicate a shared key:
  - ▶ Several DH (Diffie-Hellman) groups

# IPsec confidentiality



# IPsec confidentiality



- ▶ **DES**
  - ▶ Symmetric-key encryption, fast processing, 56-bit keys.
- ▶ **3DES**
  - ▶ Symmetric-key encryption, three independent 56-bit encryption keys per 64-bit blocks.
- ▶ **AES**
  - ▶ Stronger security than DES and faster than 3DES. Symmetric-key encryption using 128, 192 and 256-bit keys.
- ▶ **SEAL** (Software Optimized Encryption Algorithm)
  - ▶ Stream cypher with 160-bit symmetric keys

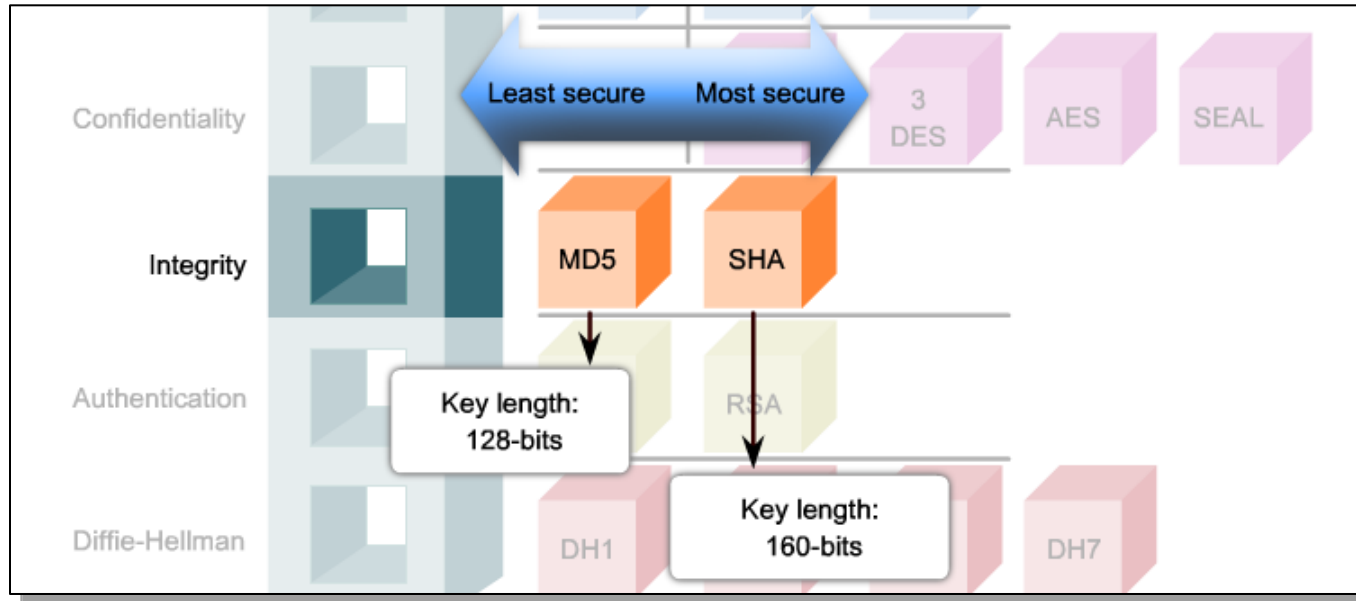


# IPsec integrity

---

- ▶ Integrity checks are required in IPsec VPNs
  - ▶ Private data is transported over public network.
  - ▶ Data can be intercepted and altered without any of the peers' knowledge.
- ▶ HMAC is a data integrity algorithm that ensures the integrity of data using hashes.
  - ▶ The sending device processes the **message** and a **shared secret key** through a hash algorithm and appends the hash to the message.
  - ▶ The receiving device recalculates the hash using the same shared key and the same algorithm and compares the hashes.

# IPsec integrity



## ▶ HMAC-Message Digest 5 (HMAC-MD5)

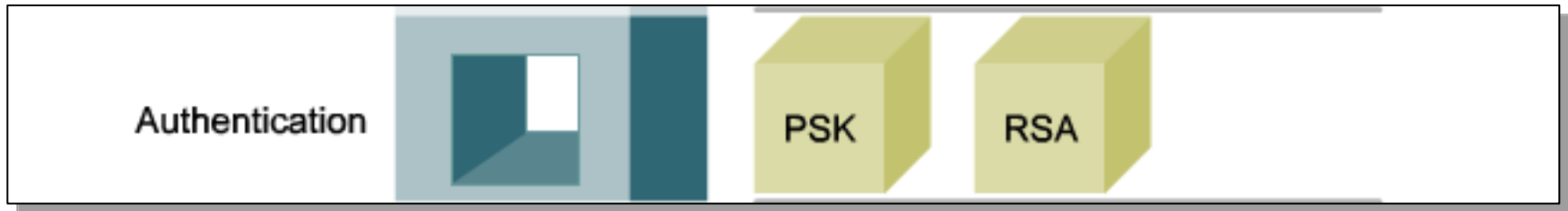
- ▶ Uses an 128-bit shared secret key to calculate an 128-bit hash.

## ▶ HMAC-Scure Hash Algorithm 1 (HMAC-SHA1)

- ▶ Uses an 160-bit secret key to calculate an 160-bit hash.

# IPsec authentication

---



- ▶ Peers must be authenticated before a communication path can be considered secure.
- ▶ Two authentication methods:
  - ▶ Pre-shared keys (PSK)
    - ▶ Must be entered in each peer, manually.
    - ▶ Easy to configure, but do not scale well.
  - ▶ RSA signatures
    - ▶ The exchange of digital certificates authenticates the peers.
    - ▶ To validate digital certificates, public/private key pairs must be used

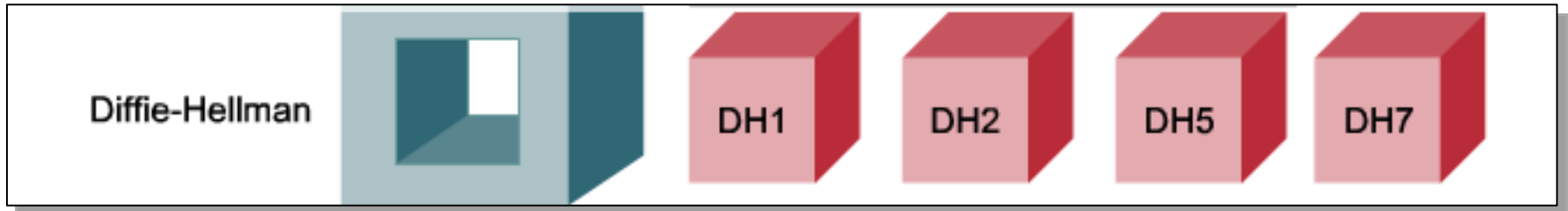
# IPsec secure key exchange

---

- ▶ Encryption algorithms such as DES, 3DES and AES require a shared secret key to perform encryption and decryption.
- ▶ MD5 and SHA-1 hashing algorithms also require secret keys to provide integrity.
- ▶ How can two devices securely communicate a secret key?
- ▶ The DH (Diffie-Hellman) key agreement is a key exchange method.
  - ▶ Allows two peers to securely communicate a secret key over an insecure channel.
- ▶ Variations of DH are called groups.

# IPsec secure key exchange

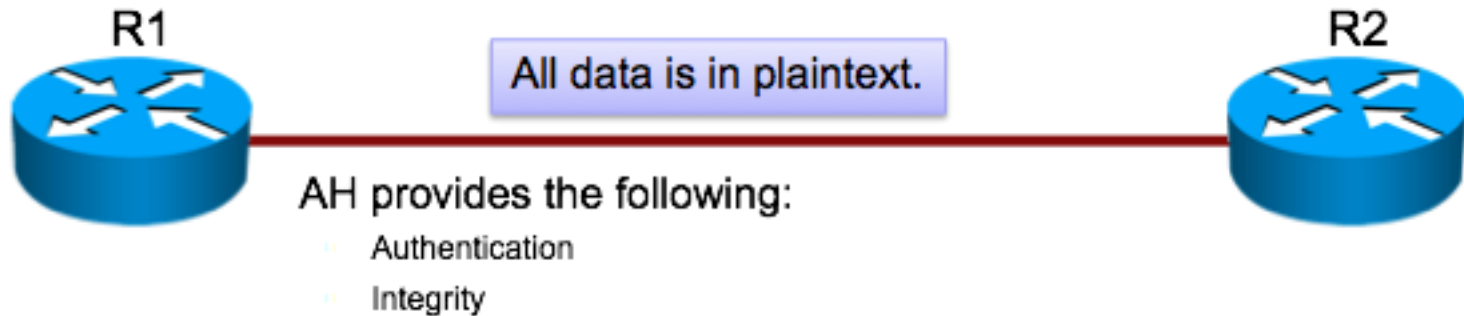
---



- ▶ There are four DH groups.
- ▶ A group specifies the length of the base prime numbers used in the algorithms (see previous lecture).
  - ▶ DH groups 1, 2 and 5 support keys of 768 bits, 1024 bits and 1536 bits, respectively.
  - ▶ AES encryption supports DH groups 2 and 5.
  - ▶ Group 7 supports Elliptical Curve Cryptography (ECC), which reduces the time needed to generate the keys.
- ▶ During the VPN tunnel setup, the devices negotiate which DH group they will use, as well as other algorithms.

# IPsec security protocols

## Authentication Header



## Encapsulating Security Payload



# Authentication Header (AH)

## AH behaviour:

1 The IP header and data payload are hashed.

IP Header + Data + Key 



Hash




Authentication Data  
(00ABCDEF)



3

The new encapsulated packet is sent to the IPsec peer router.



IP Header + Data + Key 



Hash



Recomputed Hash  
(00ABCDEF)

=

Received Hash  
(00ABCDEF)

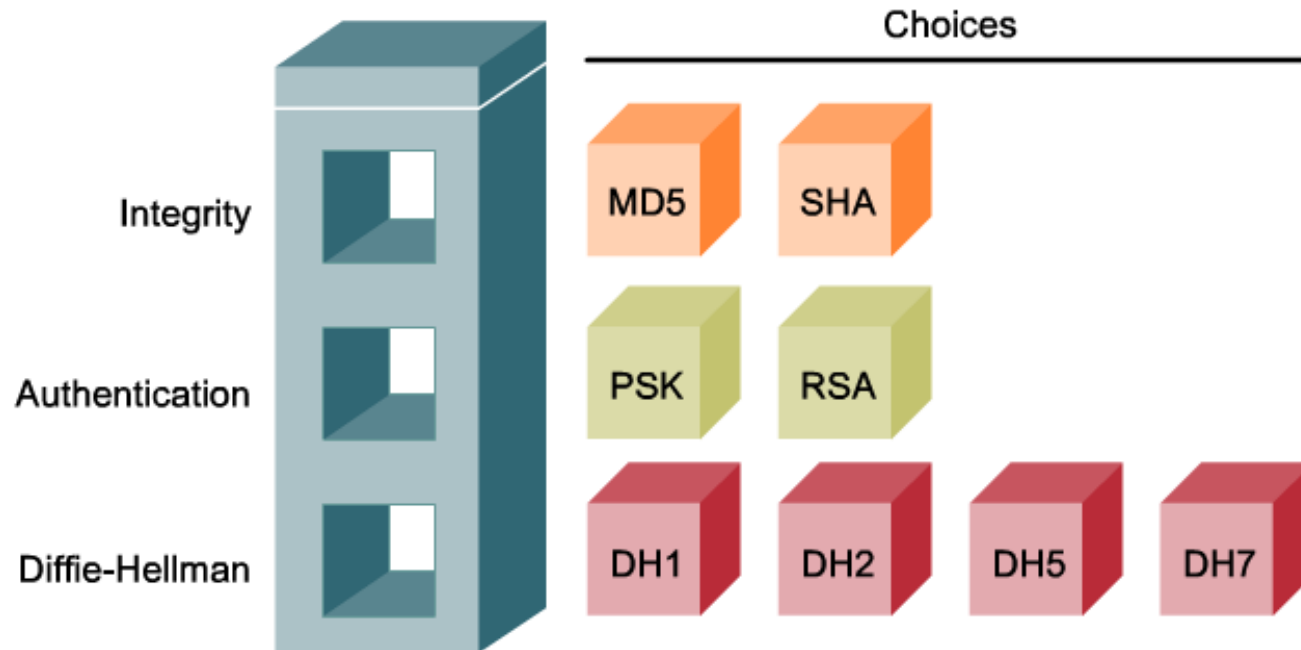
2 The hash builds an AH header that is prepended to the IP packet.

The peer router hashes the IP header and the packet payload and compares the result with the received hash.

4

# Authentication Header protocols

---



- ▶ The AH-calculated hash value does not include variable fields in the IP header (TTL).
  - ▶ NAT creates problems because AH does not expect the IP header to change.
- ▶ AH cannot provide any encryption methods: only authenticity and integrity.

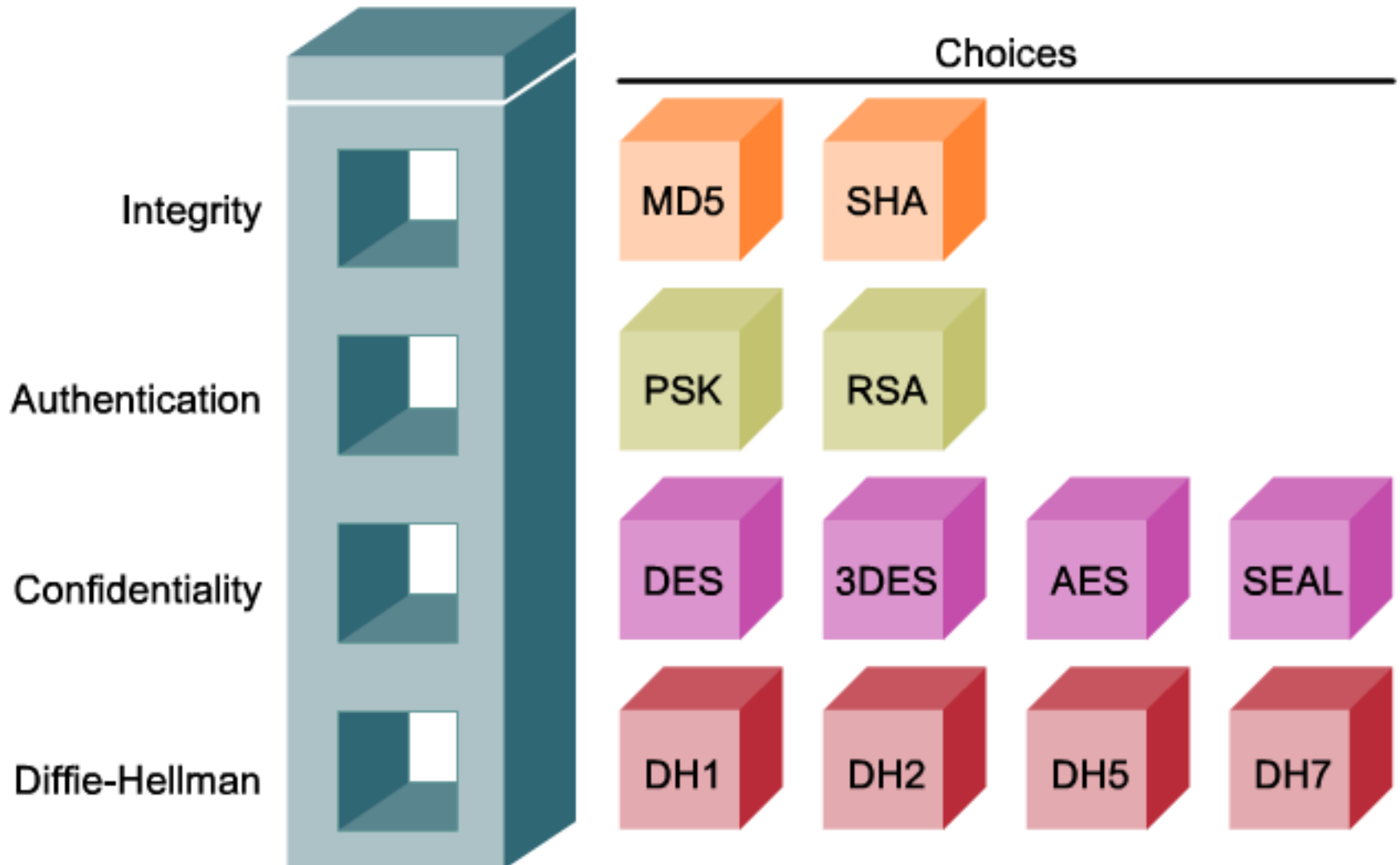


# Encapsulating Security Payload (ESP)

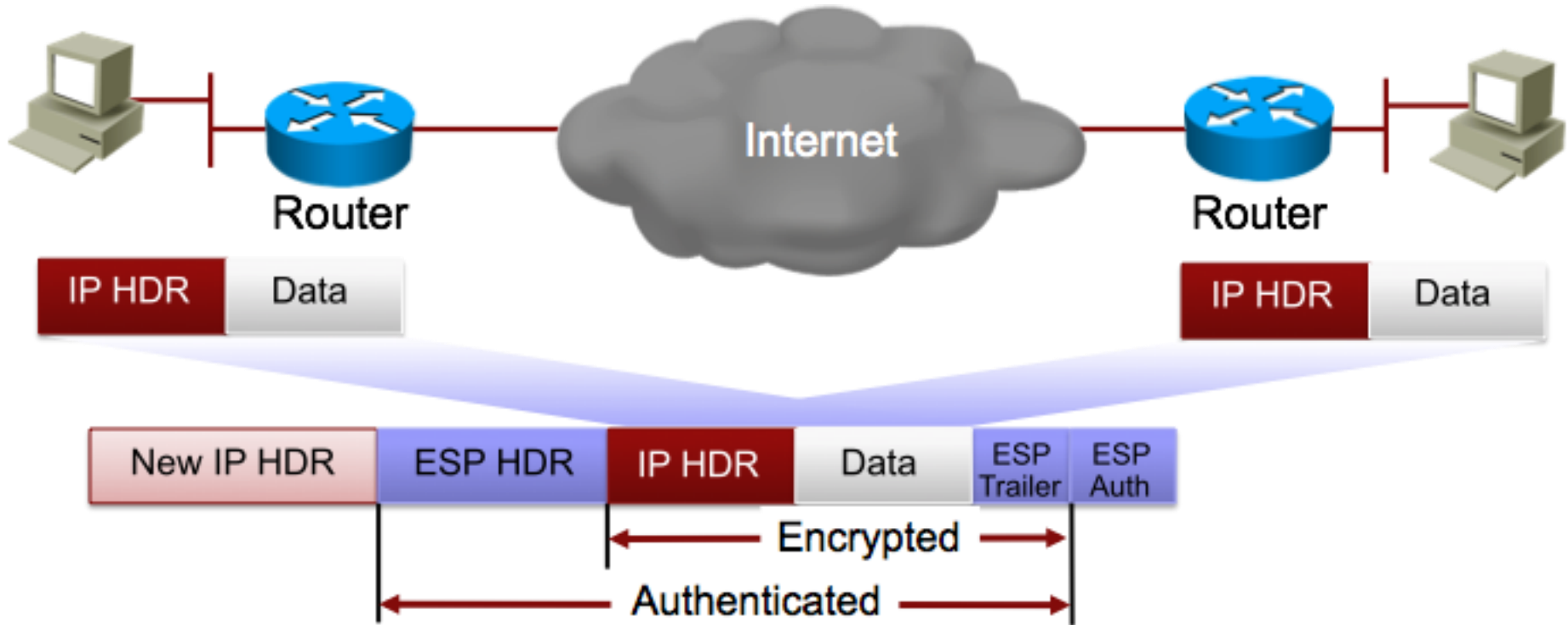
---

- ▶ ESP provides confidentiality by encrypting the payload.
- ▶ Uses a variety of symmetric-key algorithms for encryption.
  - ▶ The default is 56-bit DES.
- ▶ ESP can also provide integrity and authentication.
  - ▶ Using the same protocols and methods as AH.
- ▶ Optionally, ESP can enforce anti-replay methods.
  - ▶ Protection against duplicate packets sent from attackers.
  - ▶ Typically used in ESP, but also supported by AH.
  - ▶ How? Hash a sequence number along with the header, the packet and the secret key.

# ESP protocols



# IPsec security protocols encapsulation



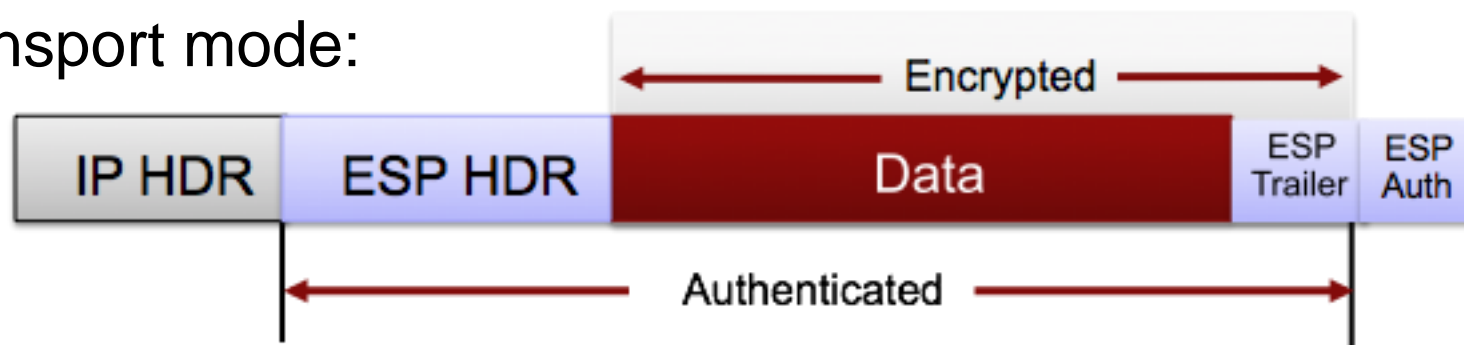
- ▶ Data is protected: the entire IP packet can be encrypted.
- ▶ Data is authenticated: the entire IP packet and the ESP header are hashed.
- ▶ The new addresses in the the new IP header are used to route the packet.
- ▶ Encryption is performed before authentication.

# IPsec encapsulation: Transport mode

---

- ▶ ESP and AH can be applied to IP packets in two different modes: **transport mode** and **tunnel mode**.

Transport mode:

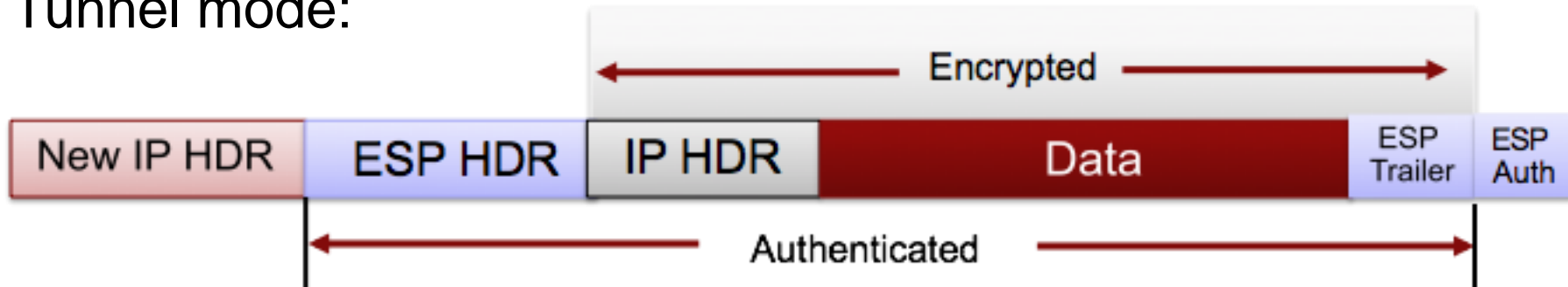


- ▶ “Transport” because security is provided only to the transport layer of the OSI model (and above, of course).
- ▶ The original IP header is left untouched (unauthenticated, unencrypted).
- ▶ Can be used with GRE (GRE hides the original IP header).
  - ▶ But can also be used when lower overhead is required.

# IPsec encapsulation: Tunnel mode

---

Tunnel mode:



- ▶ Provides security for the complete original IP packet.
- ▶ Known as “IP-in-IP” encryption.
- ▶ Can be used to extend LANs over the Internet
  - ▶ The encapsulated IP addresses can even be private.
- ▶ The packet is routed towards the destination using only the outer IP header.
  - ▶ The receiving VPN device decapsulates the packet, checks its integrity and authenticity and can route the packet further using the internal IP header.

# SA (Security Associations)

---

- ▶ VPNs negotiate security parameters, so that two peers can “talk” to each other.
- ▶ The final negotiated parameters are called a **security association (SA)**.
- ▶ SA entries are maintained in a SADB (database) and contains:
  - ▶ parameters for IPsec encryption
  - ▶ parameters for the secure key exchange
- ▶ DH is used to create the shared keys needed for encryption.
  - ▶ But the IKE protocol carries out the key exchange process.
- ▶ Keep in mind that even if public/private keys are used for authentication and key exchange, symmetric keys are still used for encryption.
  - ▶ Because they are MUCH faster.

# Internet Key Exchange (IKE)

---

- ▶ Instead of transmitting the keys directly over the network, IKE exchanges a series of data packets that allow both peers to calculate the keys.
- ▶ The exchange cannot allow a third party to deduce the key.
- ▶ IKE is defined in RFC 2409 and uses UDP port 500.
  - ▶ Hybrid protocol, combining:
    - ▶ Internet Security Association and Key Management Protocol (ISAKMP)
    - ▶ Oakley and Skeme key exchange methods
- ▶ ISAKMP defines the message format and the negotiation process carried out to establish the SAs for IPsec encryption.
- ▶ IKE is only useful as long as parameters are not configured manually.

# IKE phases

---

- ▶ The IKE protocol executes two phases to establish a secure channel:
  - ▶ Phase 1
    - ▶ The initial negotiation of SAs. The purpose of Phase 1 is to authenticate the peers and negotiate the IKE policy sets (**tunnel parameters**). A secure channel is established.
  - ▶ Phase 2
    - ▶ The secure channel already in place is used by ISAKMP to negotiate another set of SAs, this time for **encrypting traffic**. After this phase, both peers are authenticated and know the same secret key.
- ▶ After both phases have been completed, peers are ready to transfer encrypted data.



# IKE phase 1 - first exchange

---

## ▶ First exchange

- ▶ Peers negotiate the algorithms and hashes used to secure the IKE communications.
- ▶ Algorithms are grouped in IKE policy sets, which are exchanged first.
- ▶ The exchange is initiated by a proposal sent from the initiator.
- ▶ If the receiver can comply with the proposal, this proposal will be used.
- ▶ Different IKE policy sets might be needed to be configured if the peer connects to multiple peers.



## IKE phase 1 - second exchange

---

- ▶ The second exchange creates and exchanges the DH public keys between the peers.
- ▶ The DH group included in the IKE policy previously agreed upon by the peers is used.
  - ▶ This ensures that both peers use the same key generation algorithm and that they will obtain the same result.
- ▶ The key is calculated by both peers without sending the key itself.
  - ▶ See the previous course for a description of the DH algorithm.
- ▶ All further negotiations will be encrypted with the newly calculated DH secret key.

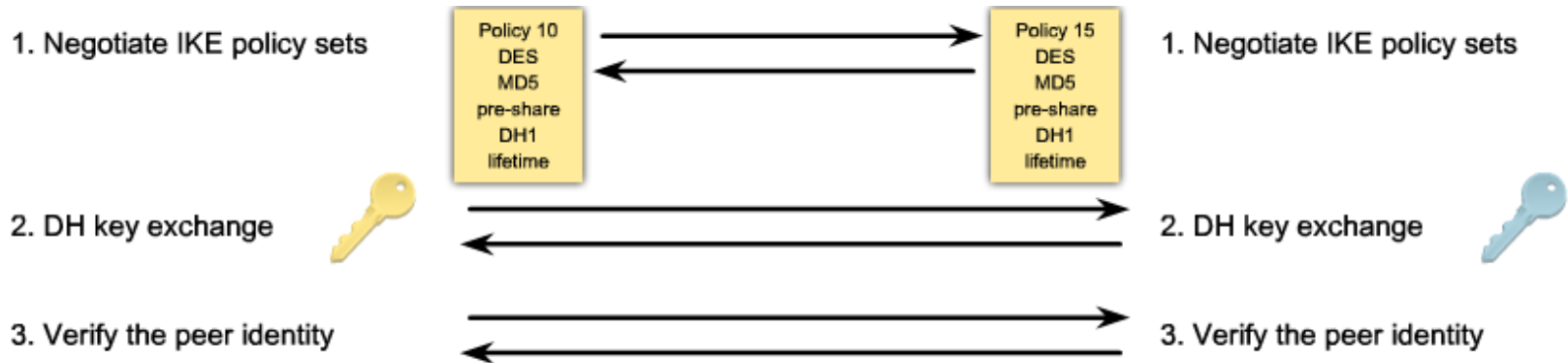
# IKE phase 1 - third exchange

---

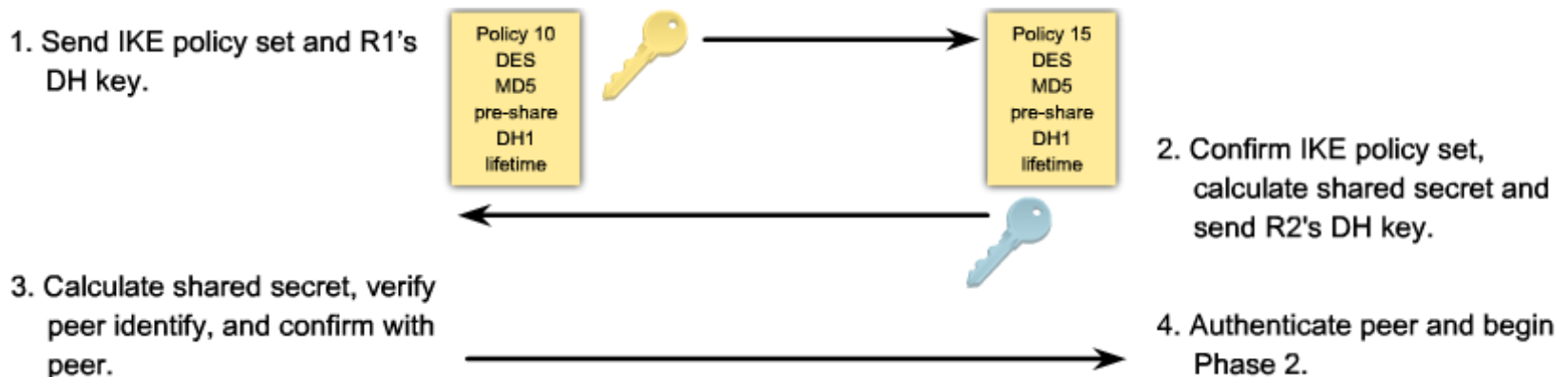
- ▶ The last exchange of the first phase authenticates the peers.
- ▶ Authentication can be carried out using:
  - ▶ a pre-shared key (PSK)
  - ▶ an RSA signature
- ▶ Peer authentication is mutual.
- ▶ After the third exchange, a bidirectional IKE SA is now established.
  
- ▶ The “three-exchange” phase 1 is called “main mode” and uses 6 packets (3 for each peer).
  - ▶ There is also an “aggressive mode” in which only 3 packets are sent.

# IKE phase 1 modes

## ▶ IKE phase 1 normal mode exchanges:



## ▶ IKE phase 1 aggressive mode exchanges:



## IKE phase 2

---

- ▶ The purpose of phase 2 is to negotiate the IPsec parameters that will be used to secure the IPsec tunnel.
- ▶ Phase 2 has only one mode
  - ▶ called “Quick mode”
- ▶ Can only occur after the initial IKE process in phase 1.
- ▶ Phase 2 negotiates another set of SAs.
  - ▶ These SAs are unidirectional
  - ▶ A separate key exchange is required for both ways.

## IKE phase 2

---

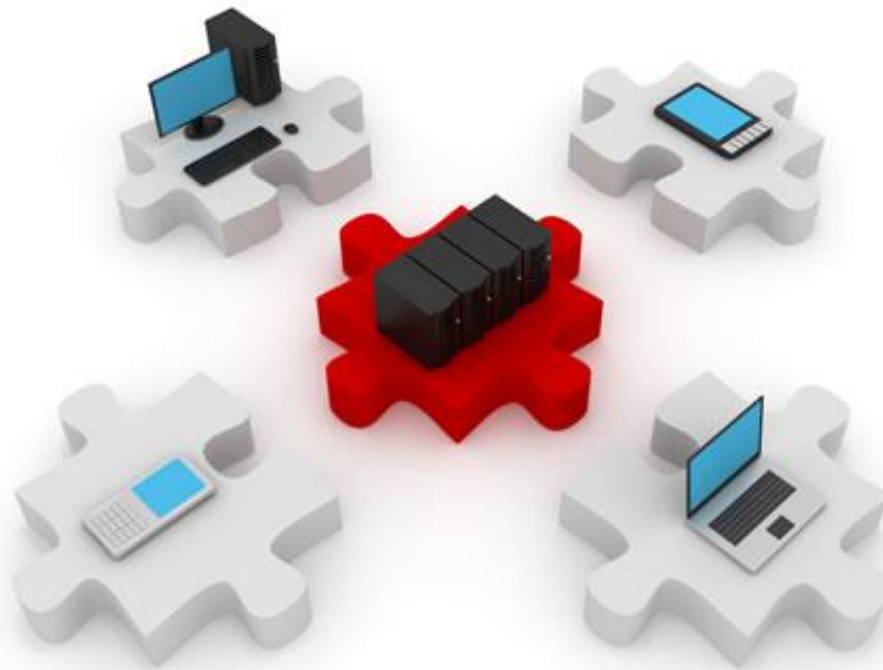


- ▶ IPsec phase 2 performs the following functions:
  - ▶ Negotiates IPsec security parameters, known as Transform Sets
  - ▶ Establishes IPsec SAs
  - ▶ Periodically regenerates IPsec SAs to ensure extra security.
  - ▶ Optionally, performs another DH exchange

# The NAT problem

---

- ▶ AH hashes the IP header and the TCP header and expects them to remain unaltered.
- ▶ NAT(PAT) overwrites the layer 3 and 4 addresses and port numbers.
- ▶ How do you solve this?
  
- ▶ **Solution: NAT-T (NAT-Traversal or NAT-Transparency)**
  - ▶ In IKE Phase 1, an unencrypted but hashed message is sent.
  - ▶ At destination, if the hashes do not match, there is a NAT router in between.
- ▶ **NAT-T encapsulates everything (including ESP) in an UDP header**
  - ▶ There is also a TCP variant available when connection state tracking is required.
    - ▶ If an IPS/IDS device is present, for example.



## Site-to-Site IPsec VPNs



# Behaviour of a VPN tunnel



1. Host A sends interesting traffic to Host B.

2. R1 and R2 negotiate an IKE Phase 1 session.



3. R1 and R2 negotiate an IKE Phase 2 session.



4. Information is exchanged via IPsec tunnel.



5. The IPsec tunnel is terminated.

# Site-to-site IPsec VPNs

---

- ▶ A VPN is a logical channel between two endpoints.
- ▶ A site-to-site VPN is a “network-to-network” virtual connection.
- ▶ VPNs do not necessarily include authentication or encryption.
  - ▶ If we need such features, we use IPsec VPNs
- ▶ A site-to-site IPsec VPN is a permanent secure virtual channel between two sites, each having one or more networks.

# Steps for configuring a site-to-site IPsec VPN

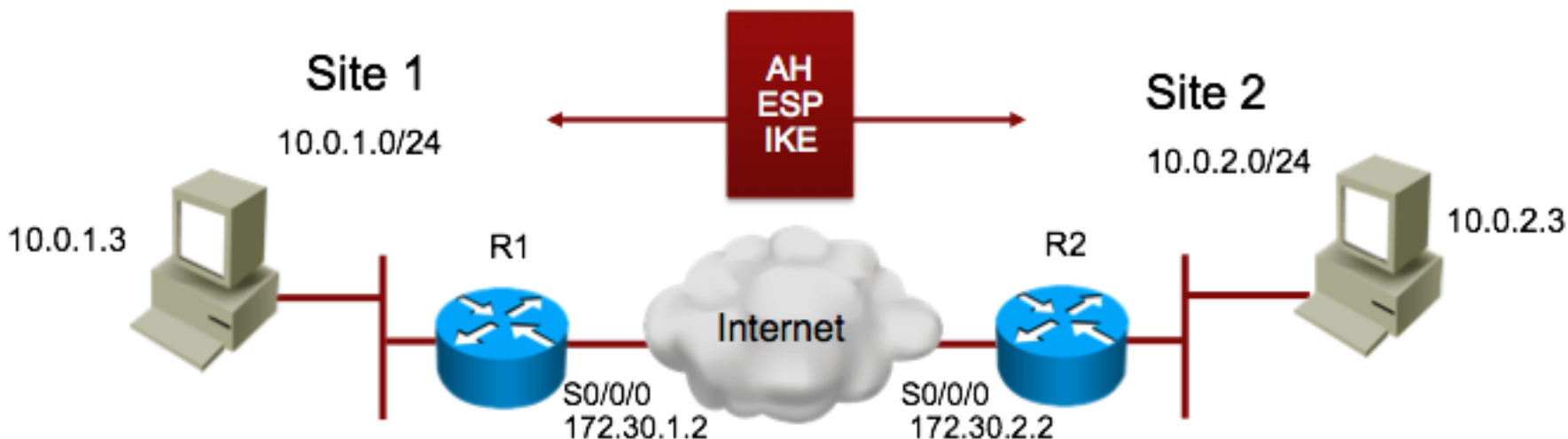
---

## Tasks to Configure IPsec:

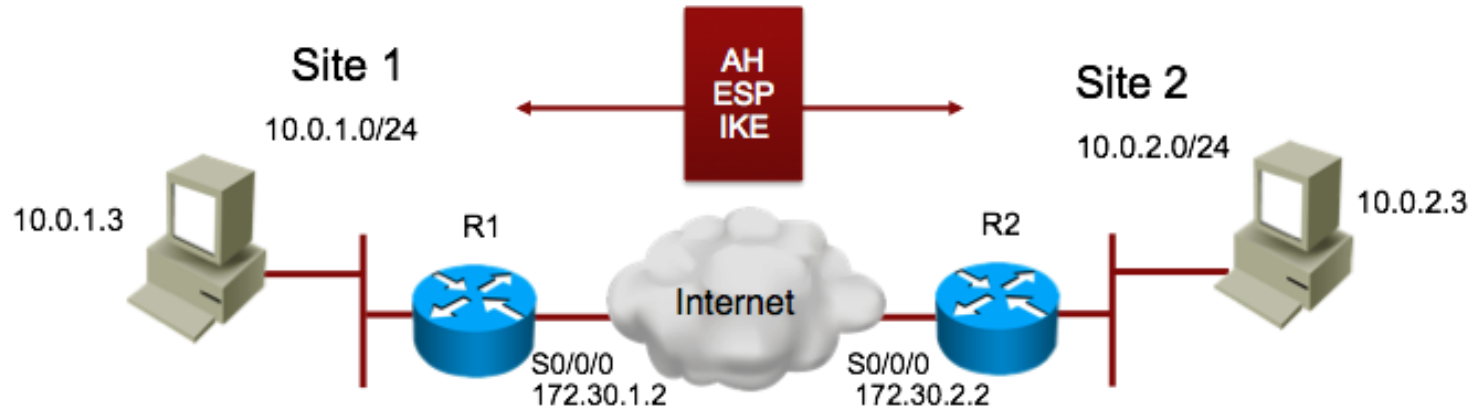
- Task 1: Ensure that ACLs are compatible with IPsec.
- Task 2: Create ISAKMP (IKE) policy.
- Task 3: Configure IPsec transform set.
- Task 4: Create a crypto ACL.
- Task 5: Create and apply the crypto map.

# 1. ACL configuration

- ▶ Ensure that existing ACLs do not block IPsec and/or IKE traffic.
  - ▶ AH is IP protocol number 51
  - ▶ ESP is IP protocol number 50
  - ▶ IKE uses UDP port number 500
- ▶ Don't confuse protocol numbers with port numbers!

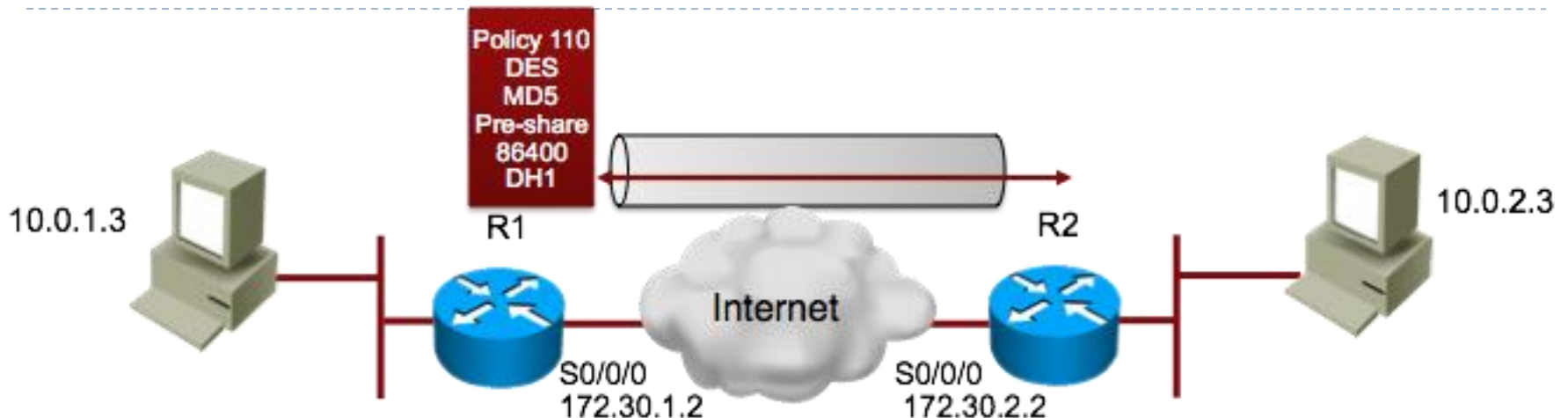


# 1. ACL configuration example



```
R1(config)# access-list 102 permit ahp host 172.30.2.2 host 172.30.1.2
R1(config)# access-list 102 permit esp host 172.30.2.2 host 172.30.1.2
R1(config)# access-list 102 permit udp host 172.30.2.2 host 172.30.1.2 eq isakmp
R1(config)#
R1(config)# interface Serial0/0/0
R1(config-if)# ip address 172.30.1.2 255.255.255.0
R1(config-if)# ip access-group 102 in
!
R1(config)# exit
R1# 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 isakmp
```

## 2. IKE policy configuration



- ▶ These parameters are used during negotiation.
- ▶ The negotiation's purpose is to establish an ISAKMP peering between two IPsec endpoints.
- ▶ Multiple ISAKMP policies can be configured, each with a unique priority number (1 to 10000).
- ▶ More secure policies should have lower priority numbers.
  - ▶ The security association will be made up from the lowest common set of policy that both peers agree upon.
- ▶ If commands are not explicitly entered, defaults are used
  - ▶ For example, if no hash algorithm is set, it defaults to SHA.

## 2. IKE policy configuration

---

- ▶ Enter ISAKMP policy configuration mode:

```
R1(config)#crypto isakmp policy 110
```

- ▶ Set the ISAKMP parameters

```
R1(config-isakmp)#authentication pre-share
```

```
R1(config-isakmp)#encryption des
```

```
R1(config-isakmp)#group 1
```

```
R1(config-isakmp)#hash md5
```

```
R1(config-isakmp)#lifetime 86400
```

- ▶ Policy numbers are locally significant only (110).
- ▶ Policy numbers don't need to match, only their contents
  - ▶ All parameters must be equal for a policy to match
  - ▶ Except for the key lifetime (the lowest lifetime is still accepted)

## 2. IKE policy parameters

---

ISAKMP Parameters				
Parameter	Keyword	Accepted Values	Default Value	Description
encryption	des 3des aes aes 192 aes 256	56-bit Data Encryption Standard Triple DES 128-bit AES 192-bit AES 256-bit AES	des	Message encryption algorithm
hash	sha md5	SHA-1 (HMAC variant) MD5 (HMAC variant)	sha	Message integrity (Hash) algorithm
authentication	pre-share rsa-encr rsa-sig	preshared keys RSA encrypted nonces RSA signatures	rsa-sig	Peer authentication method
group	1 2 5	768-bit Diffie-Hellman (DH) 1024-bit DH 1536-bit DH	1	Key exchange parameters (DH group identifier)
lifetime	<i>seconds</i>	Can specify any number of seconds	86,400 sec (one day)	ISAKMP-established SA lifetime

Note: Actual parameters vary based on IOS image.



## 2. Negotiating multiple policies



R1(config)#

```
crypto isakmp policy 100
  hash md5
  authentication pre-share
!
crypto isakmp policy 200
  hash sha
  authentication rsa-sig
!
crypto isakmp policy 300
  hash md5
  authentication pre-share
```

R2(config)#

```
crypto isakmp policy 100
  hash md5
  authentication pre-share
!
crypto isakmp policy 200
  hash sha
  authentication rsa-sig
!
crypto isakmp policy 300
  hash md5
  authentication rsa-sig
```

- ▶ In this example, policies 100 and 200 can be negotiated, but 300 cannot.
- ▶ A peer sends all its policies to the remote peer.
- ▶ The remote peer tries to find a complete match with its own policies.

## 2. IKE configuration: pre-shared keys

---

- ▶ If pre-shared keys have been specified (and negotiated) in the IKE policy, then a key must be configured.
- ▶ The PSK must be identical for two peers to create an SA.
- ▶ Different PSKs can be configured for different peering relationships.
  
- ▶ Example:

```
R1(config)#crypto isakmp key cisco123 address 172.30.2.2  
[similar config for R2, using R1's IP address]
```
- ▶ Note: a hostname can be used instead of the IP address, but it will only be sent as an identity indicator. To configure a valid peering, you must specify the remote peer's IP address.

## 3. Transform set configuration

---

- ▶ A transform set is a set of protocol-algorithm pairs designed to protect the data flow through the tunnel.
- ▶ Each protocol-algorithm pair is called a “transform”.
- ▶ A transform set can have up to 4 transforms
  - ▶ One AH authentication method
  - ▶ One ESP encryption method
  - ▶ One ESP authentication method
  - ▶ One compression method
- ▶ During the negotiation, the peers search for a transform set that matches between both peers.
- ▶ If ISAKMP is not used to establish SAs, only one, non-negotiated transform set will be used.

### 3. Allowed transform combinations

---

- ▶ AH transform:
  - ▶ ah-md5-hmac; ah-sha-hmac
- ▶ ESP encryption transform:
  - ▶ esp-aes; esp-aes 192; esp-aes 256
  - ▶ esp-des; esp-3des
  - ▶ esp-seal; esp-null
- ▶ ESP authentication transform:
  - ▶ esp-md5-hmac
  - ▶ esp-sha-hmac
- ▶ IP compression transform:
  - ▶ comp-lzs

# 3. Transform set example

---

## ▶ Sample transform sets:

```
R(config)#crypto ipsec transform-set RED ah-md5-hmac esp-3des  
esp-md5-hmac comp-lzs
```

- ❑ Uses AH with HMAC authentication
- ❑ Uses ESP with both MD5 authentication and 3DES encryption
- ❑ Uses IP header encryption with the LZS algorithm

```
R(config)#crypto ipsec transform-set YELLOW ah-md5-hmac esp-aes
```

- ❑ Uses AH with MD5 authentication
- ❑ Uses ESP with AES encryption

```
R(config)#crypto ipsec transform-set BLUE esp-aes esp-sha-hmac
```

- ❑ Uses both ESP AES encryption and ESP SHA authentication.

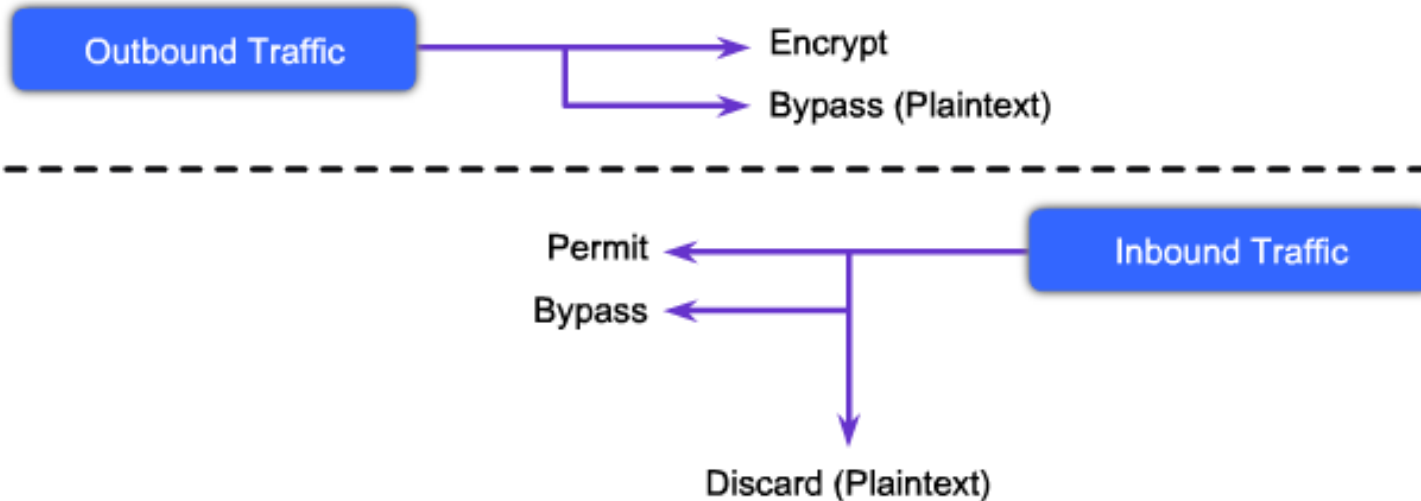
## 4. Crypto ACL configuration

---

- ▶ Crypto ACL are just simple ACLs that select traffic flows to protect.
- ▶ On an outbound crypto ACL:
  - ▶ a permit statement indicates that the traffic must be encrypted
  - ▶ a deny statement indicates that the traffic must be sent in clear text
  - ▶ traffic is not dropped by a deny statement in a crypto ACL
- ▶ On an inbound crypto ACL:
  - ▶ a permit statement must match incoming encrypted traffic
  - ▶ a deny statement must match incoming clear text traffic
  - ▶ inbound ACL are used to discard traffic that should have been protected by IPsec.

## 4. Crypto ACLs

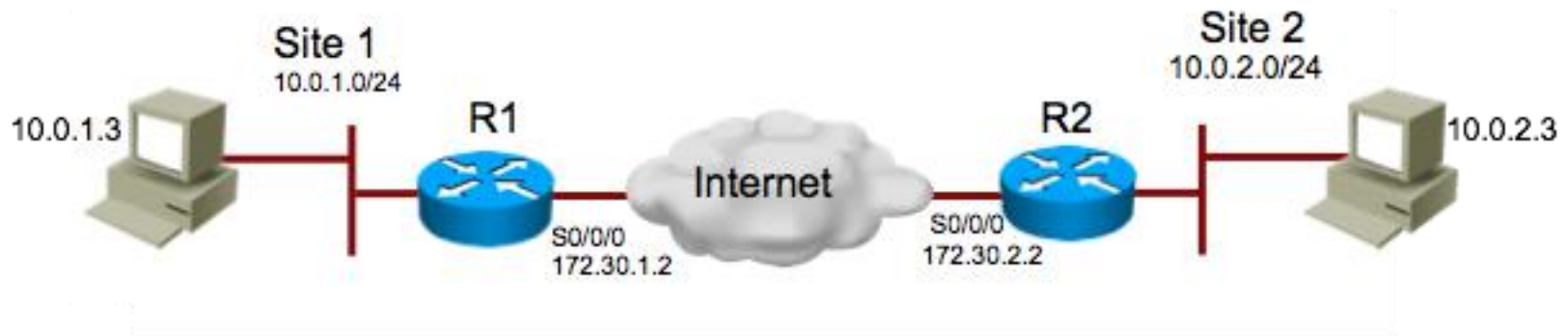
---



- ▶ Outbound indicates the data flows to be protected by the ACL.
- ▶ Inbound filters traffic that should have been protected by the ACL.
- ▶ Crypto ACLs must be extended ACLs.

## 4. Crypto ACLs sample configuration

---



- ▶ Applied to R1 S0/0/0 outbound traffic:

```
R1(config)# access-list 110 permit tcp 10.0.1.0 0.0.0.255  
10.0.2.0 0.0.0.255
```

- ▶ Applied to R2 S0/0/0 outbound traffic:

```
R2(config)# access-list 101 permit tcp 10.0.2.0 0.0.0.255  
10.0.1.0 0.0.0.255
```

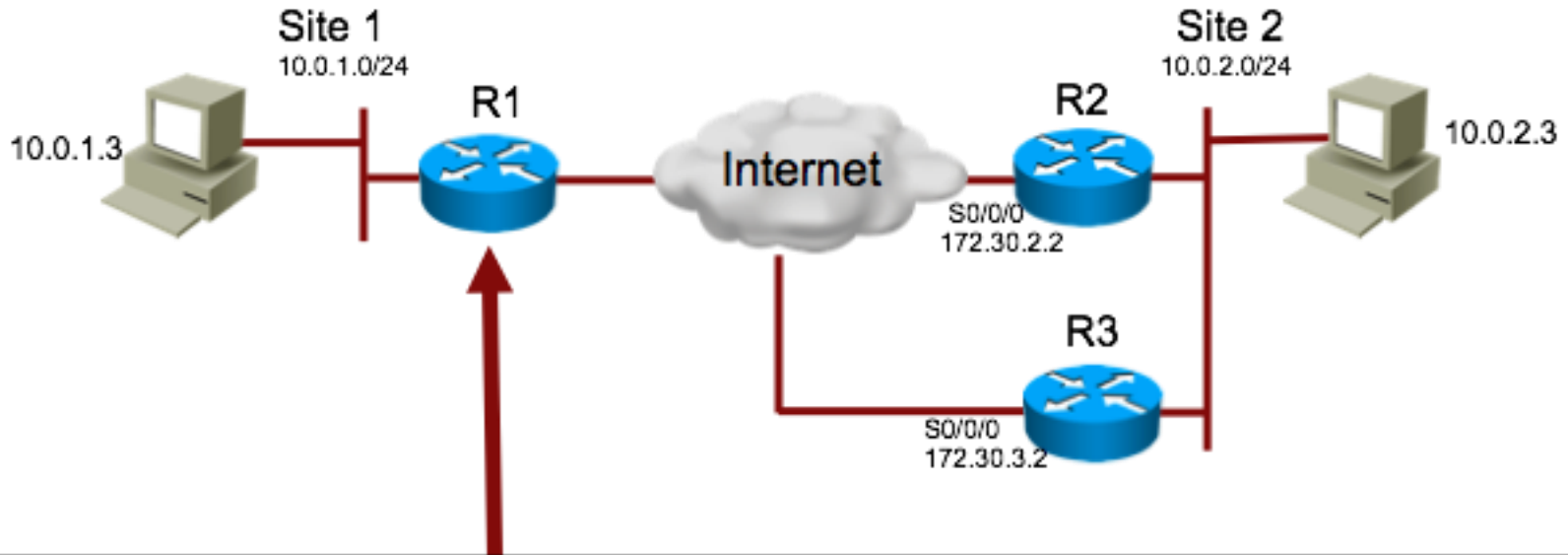


## 5. The Crypto map

---

- ▶ A crypto map is a “sum” of all configuration items that we’ve discussed so far.
- ▶ It combines the following parameters:
  - ▶ The crypto ACL that indicates which traffic to protect
  - ▶ The security associations used to establish the tunnel
  - ▶ Who is the remote peer
  - ▶ Which local address will be used for IPsec traffic (optional)
  - ▶ Which transform sets will be used in negotiation for data protection.
- ▶ Crypto maps have names and sequence numbers.
  - ▶ Maps with the same name and different sequence numbers are grouped in a crypto map set.
- ▶ Only one crypto map set can be assigned to an interface.
- ▶ Multiple interfaces can share the same crypto map.

## 5. Sample crypto map configuration



```
R1(config)# crypto map MYMAP 10 ipsec-isakmp  
R1(config-crypto-map)# match address 110  
R1(config-crypto-map)# set peer 172.30.2.2 default  
R1(config-crypto-map)# set peer 172.30.3.2  
R1(config-crypto-map)# set pfs group1  
R1(config-crypto-map)# set transform-set mine  
R1(config-crypto-map)# set security-association lifetime seconds 86400
```

- ▶ Multiple peers can be specified for redundancy.

## 5. Crypto map configuration explained

---

```
R1(config)# crypto map MYMAP 10 ipsec-isakmp
```

- ▶ Name of crypto map and sequence number

```
R1(config-crypto-map)# match address 110
```

- ▶ Crypto ACL to be matched for encrypted traffic.

```
R1(config-crypto-map)# set peer 172.30.2.2 default
```

- ▶ Configure the primary peer

```
R1(config-crypto-map)# set peer 172.30.3.2
```

- ▶ Optionally, configure a secondary peer.

```
R1(config-crypto-map)# set pfs group1
```

- ▶ Set DH group 1.

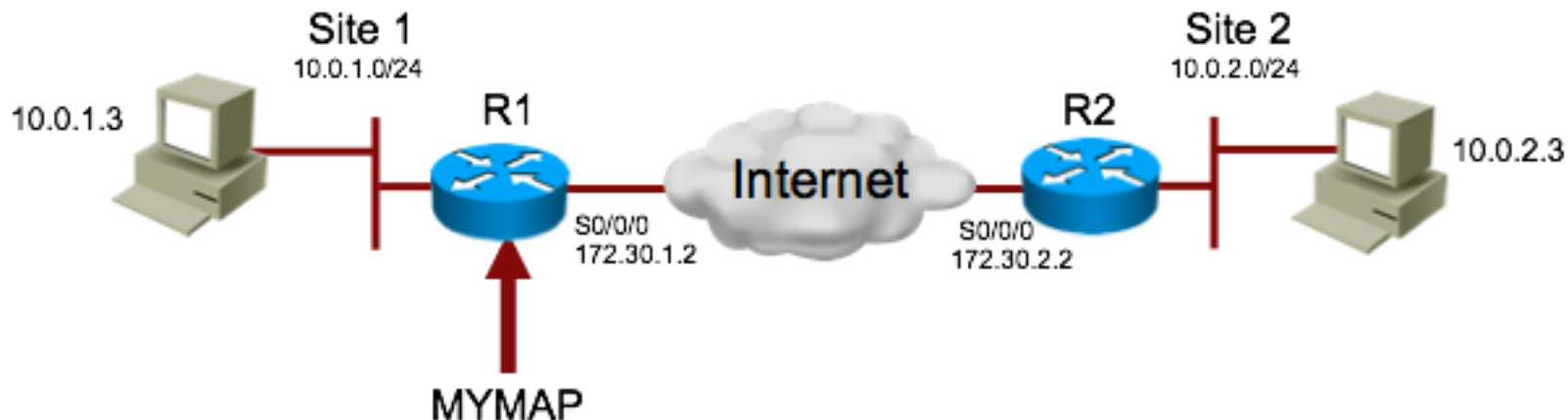
```
R1(config-crypto-map)# set transform-set mine
```

- ▶ Chose an already-configured transform set to use in ISAKMP negotiation.

```
R1(config-crypto-map)# set security-association lifetime seconds  
86400
```

- ▶ Set an already-configured transform set to use in the IPsec tunnel.

# FINALLY!!! Assigning the crypto map



```
R1(config)# interface serial0/0/0  
R1(config-if)# crypto map MYMAP
```

- ▶ Applies the crypto map to the outgoing interface.
- ▶ Activates the IPsec policy.
- ▶ Aaand... you're done!

# Verifying and troubleshooting IPsec

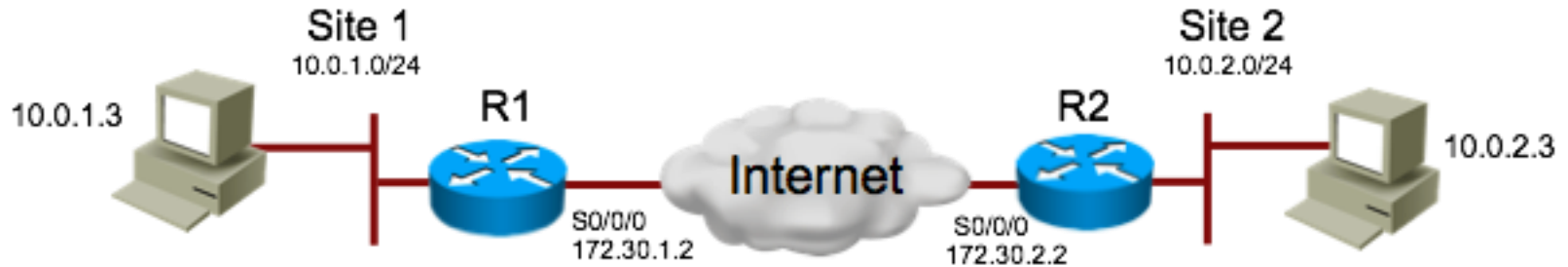
---

<b>Show command</b>	<b>Description</b>
<code>show crypto map</code>	Displays configured crypto maps.
<code>show crypto isakmp policy</code>	Displays configured IKE policies.
<code>show crypto ipsec sa</code>	Displays established IPsec tunnels.
<code>show crypto ipsec transform-set</code>	Displays configured IPsec transform sets.
<code>debug crypto isakmp</code>	Debug IKE events.
<code>debug crypto ipsec</code>	Debug IPsec events.

---

# The “show crypto map” command

---



- ▶ Displays the currently configured crypto maps.

```
R1# show crypto map
```

```
Crypto Map "MYMAP" 10 ipsec-isakmp
```

```
Peer = 172.30.2.2
```

```
Extended IP access list 110
```

```
access-list 102 permit ip host 10.0.1.3 host 10.0.2.3
```

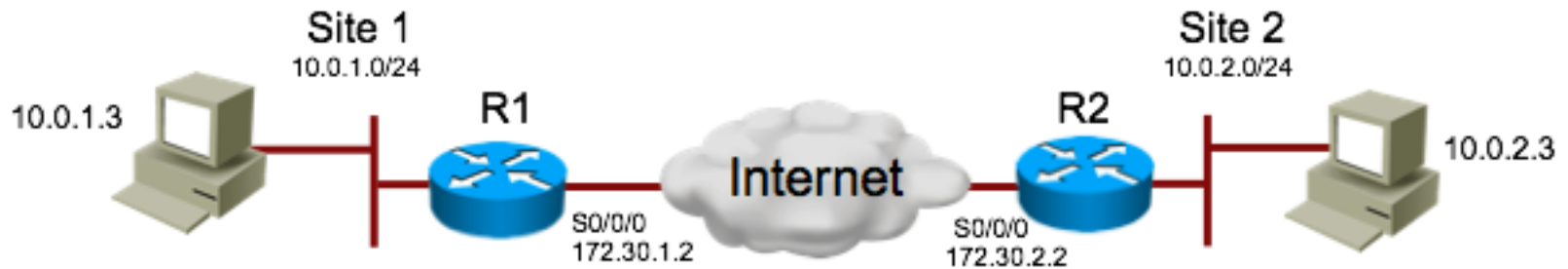
```
Current peer: 172.30.2.2
```

```
Security association lifetime: 4608000 kilobytes/3600 seconds
```

```
PFS (Y/N): N
```

```
Transform sets={ MYSET, }
```

# The “show crypto isakmp policy” command



- ▶ Displays configured IKE policies:

```
R1# show crypto isakmp policy
```

```
Protection suite of priority 110
```

```
  encryption algorithm: 3DES - Data Encryption Standard (168 bit keys).
  hash algorithm:       Secure Hash Standard
  authentication method: preshared
  Diffie-Hellman group: #2 (1024 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
```

# The “show crypto ipsec transform-set” command

---



- ▶ Displays configured IPsec transform sets:

```
R1# show crypto ipsec transform-set
```

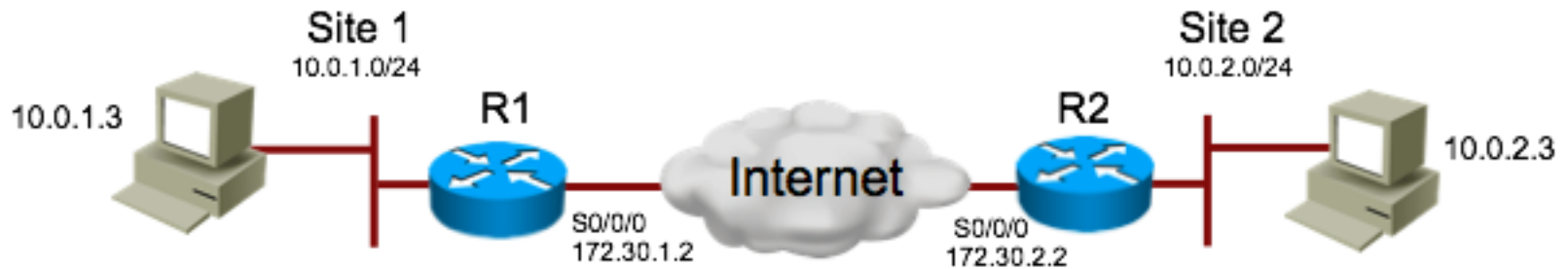
```
Transform set AES_SHA: { esp-128-aes esp-sha-hmac }
```

```
will negotiate = { Tunnel, },
```



# The “show crypto ipsec sa” command

---



- ▶ Displays established IPsec tunnels

```
R1# show crypto ipsec sa
```

```
Interface: Serial10/0/0
```

```
  Crypto map tag: MYMAP, local addr. 172.30.1.2
```

```
  local ident (addr/mask/prot/port): (172.30.1.2/255.255.255.255/0/0)
```

```
  remote ident (addr/mask/prot/port): (172.30.2.2/255.255.255.255/0/0)
```

```
  current_peer: 172.30.2.2
```

```
    PERMIT, flags={origin_is_acl,}
```

```
  #pkts encaps: 21, #pkts encrypt: 21, #pkts digest 0
```

```
  #pkts decaps: 21, #pkts decrypt: 21, #pkts verify 0
```

```
  #send errors 0, #recv errors 0
```

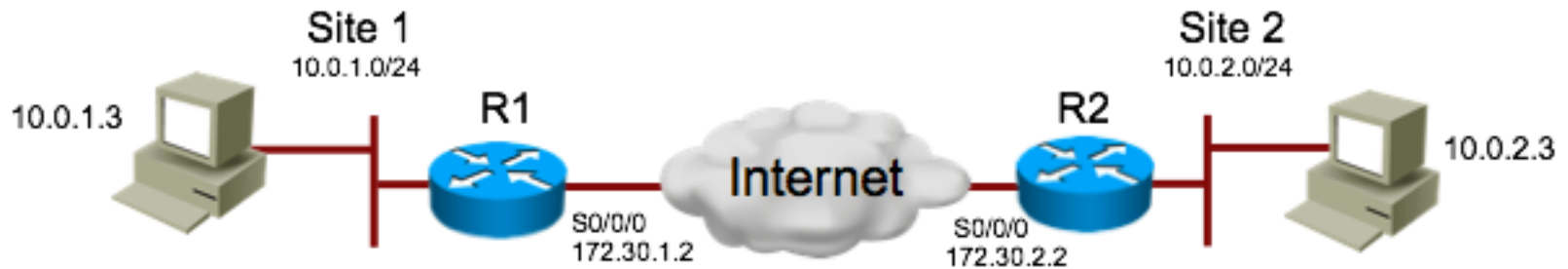
```
  local crypto endpt.: 172.30.1.2, remote crypto endpt.: 172.30.2.2
```

```
  path mtu 1500, media mtu 1500
```

```
  current outbound spi: 8AE1C9C
```

# Simple debugging example

---



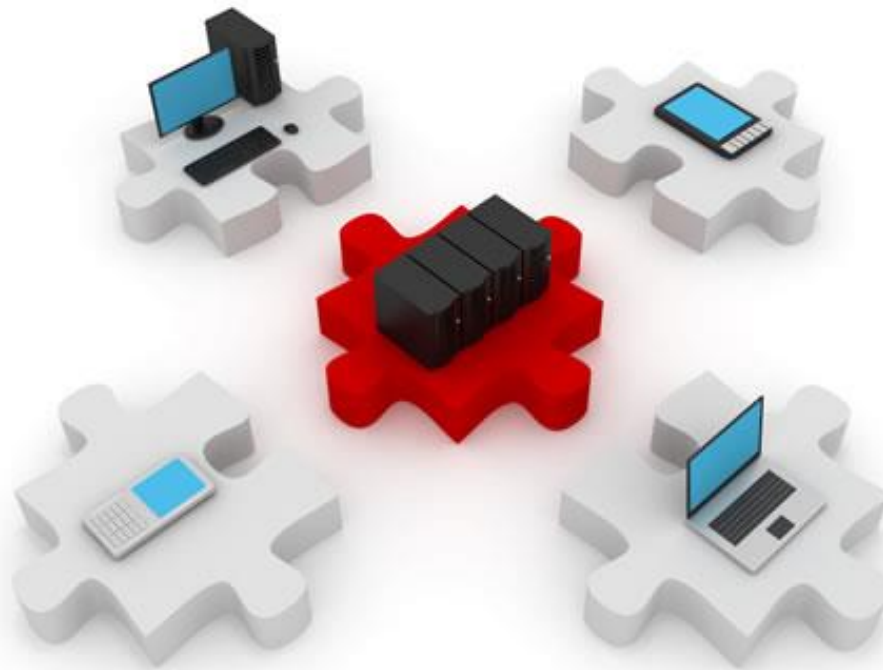
```
R1#debug crypto isakmp
```

```
1d00h: ISAKMP (0:1): atts are not acceptable. Next payload is 0 1d00h:  
ISAKMP (0:1); no offers accepted!
```

```
1d00h: ISAKMP (0:1): SA not acceptable!
```

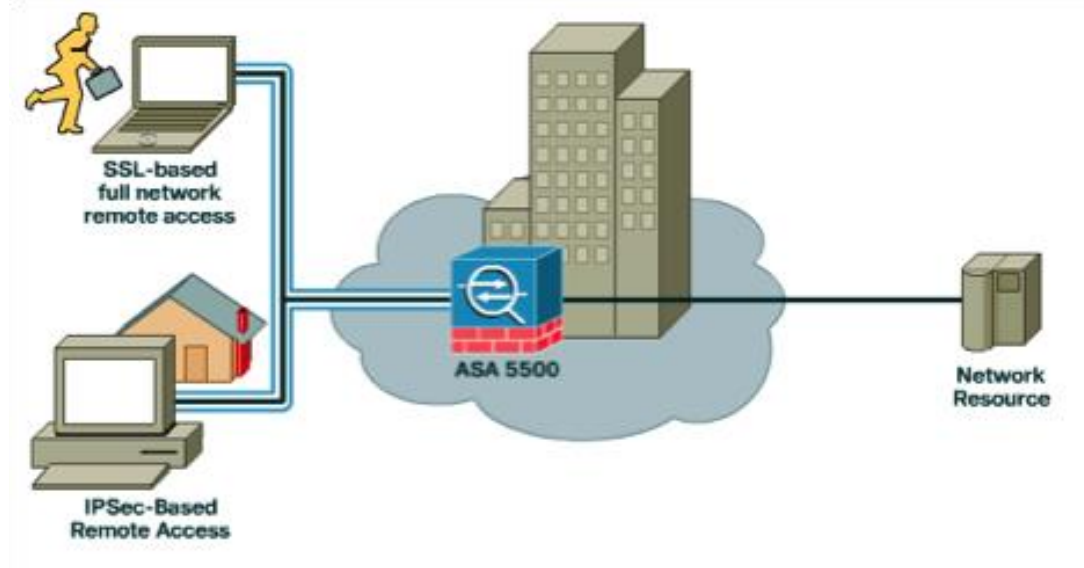
```
1d00h: %CRYPTO-6-IKMP_MODE_FAILURE: Processing of Main Mode failed with  
peer at 172.30.2.2
```

- ▶ What does the above debug message indicate?
  - ▶ The error has occurred in main mode (IKE phase 1)
  - ▶ The phase 1 policy did not negotiate successfully
- ▶ Check that there is a corresponding IKE policy on both sides, that can match between the peers.



# Remote-Access

# Remote access



- ▶ VPNs offer remote access to a local network's internal resources.
- ▶ VPNs can be gateways towards any destination, for any application.
- ▶ Two methods for remote-access VPN: IPsec and SSL VPNs

**IPsec Remote  
Access VPN**

Any  
Application

Anywhere  
Access

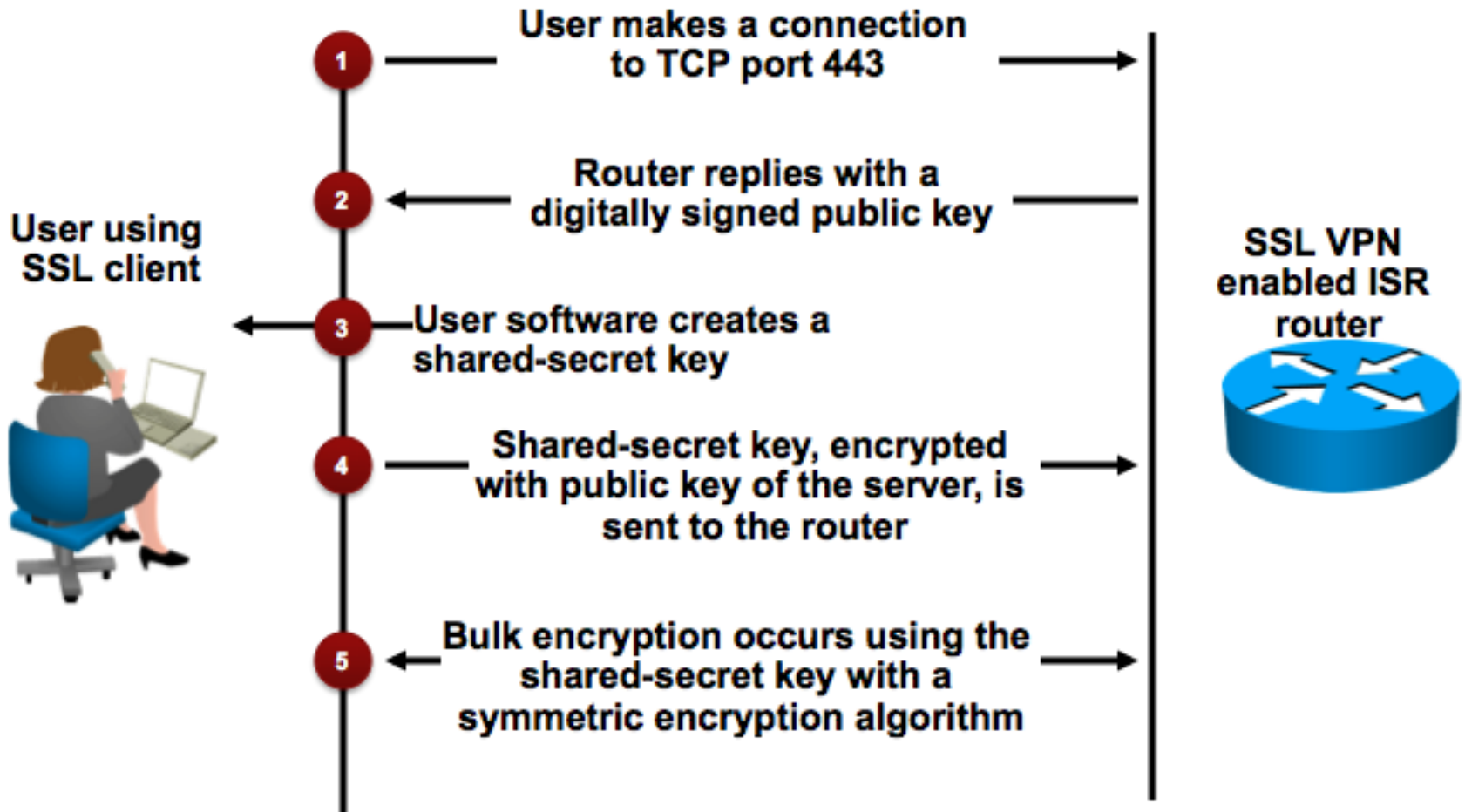
**SSL-Based  
VPN**

# Comparison of SSL and IPsec VPNs

	SSL	IPsec
Applications	Web-enabled applications, file-sharing, e-mail	All IP-based applications
Encryption	<b>Moderate</b> Key lengths from 40 bits to 128 bits	<b>Stronger</b> Key lengths from 56 bits to 256 bits
Authentication	<b>Moderate</b> One-way or two-way authentication	<b>Strong</b> Two-way authentication using shared secrets or digital certificates
Ease of use	<b>Very high</b>	<b>Moderate</b> Can be challenging to non-technical users
Overall security	<b>Moderate</b> Anyone can connect	<b>Strong</b> Only specific devices with specific configurations can connect.

# Establishing an SSL session

---



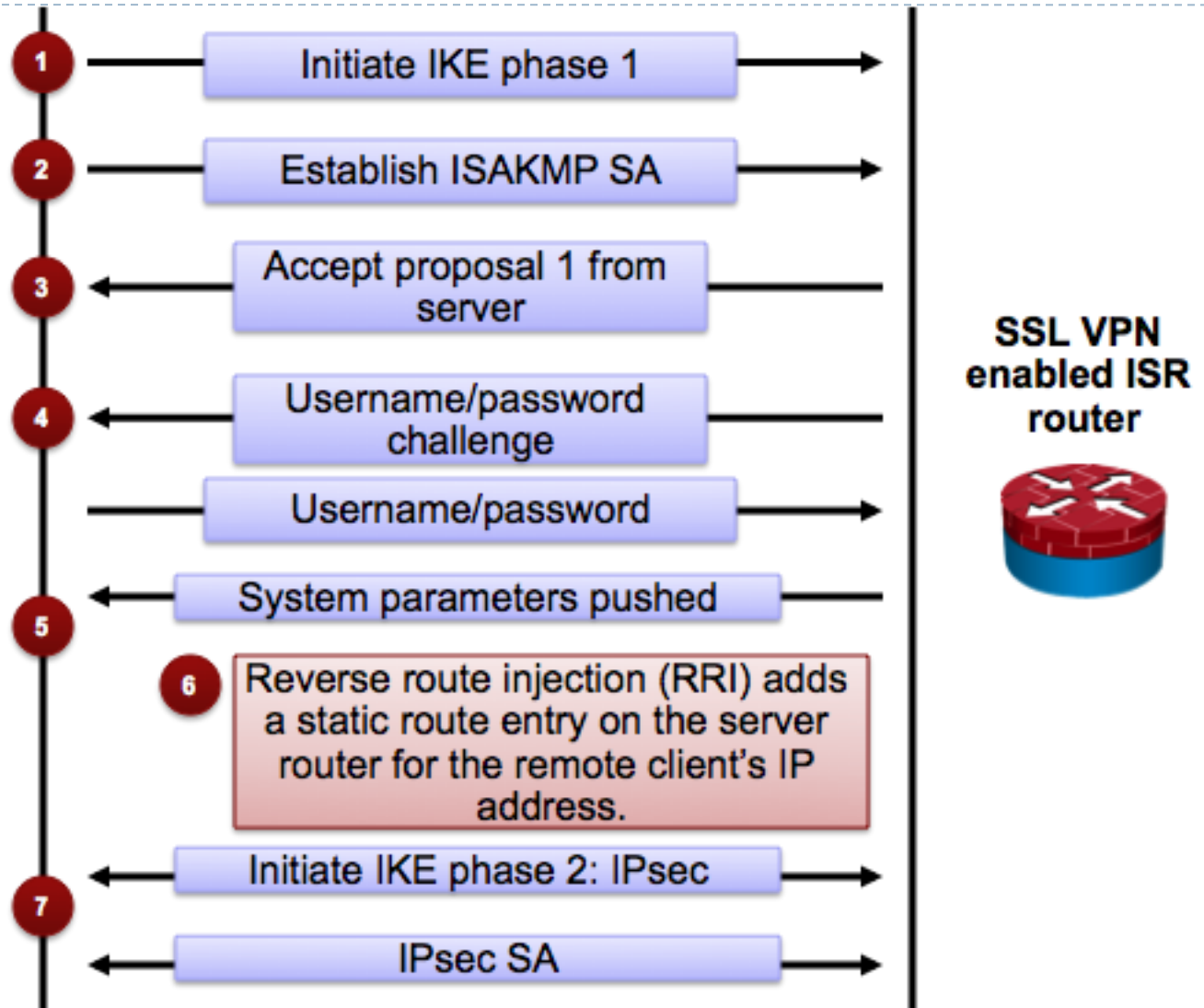
# Cisco Easy VPN

---

- ▶ Simplified deployment method for the client side.
- ▶ All VPN parameters are defined on the Cisco IOS Easy VPN Server
- ▶ All security policies are pushed to connecting clients.
  - ▶ Client configuration is minimized
  - ▶ Data security is automatically secured, with minimal client interaction.
- ▶ Cisco Easy VPN automates:
  - ▶ tunnel parameter negotiation (addresses, algorithms, lifetimes, etc)
  - ▶ tunnel establishment
  - ▶ NAT/PAT and ACL creation
  - ▶ user authentication
  - ▶ security key management
  - ▶ authentication, encryption and decryption of data packets

# Cisco Easy VPN session establishment

User using  
SSL client



SSL VPN  
enabled ISR  
router





# Summary and questions

---

- ▶ Types of VPNs: site-to-site and remote-access.
- ▶ The GRE tunnel mode.
- ▶ Algorithms behind IPsec VPNs.
- ▶ Configuring a site-to-site IPsec VPN.
- ▶ Remote access and SSL VPNs.

# Paranoia

*"Just because you're paranoid doesn't mean that there isn't someone out there ready to get you..."*

***Anonymous***