CCS6224
Network Security

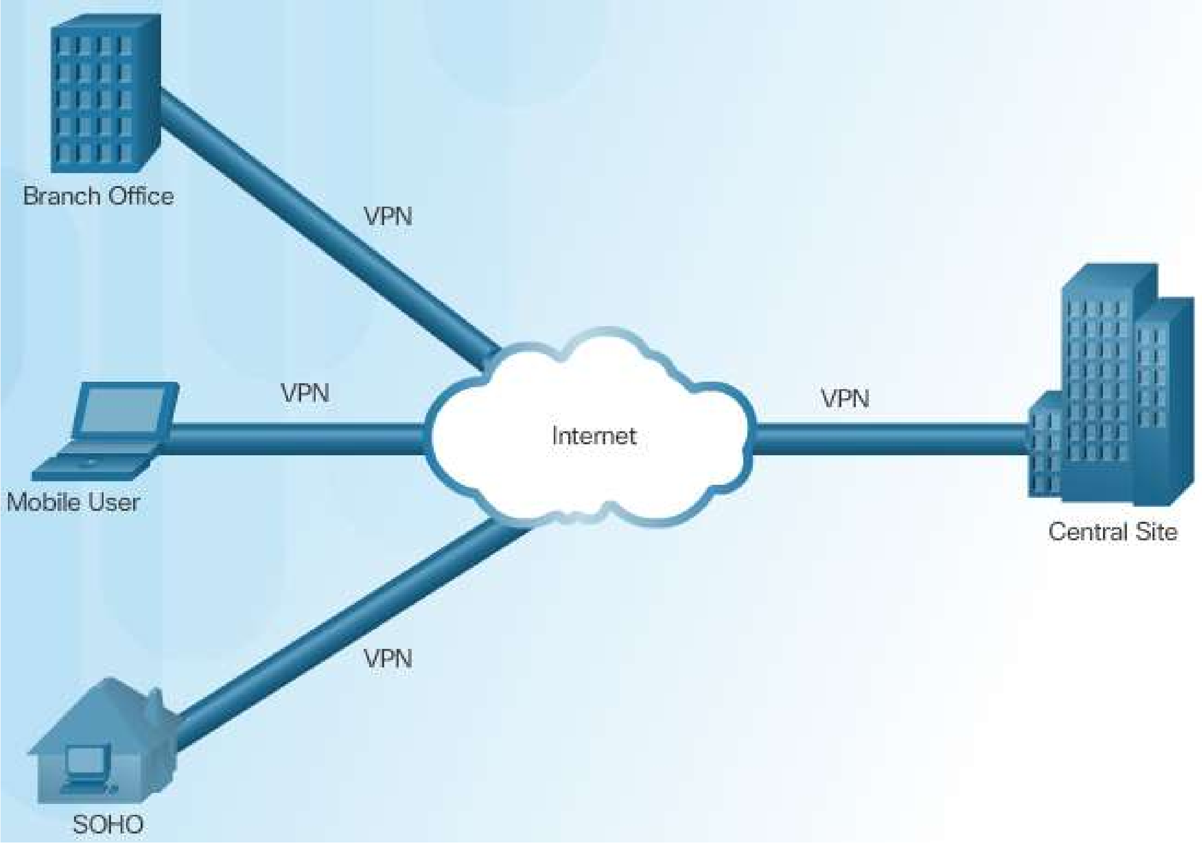
Lecture 5
Virtual Private Network (VPN)

Outline

- Introduction to VPN
- IPsec VPN Components and Operations
- Implementing Site-to-Site IPsec VPNs

Introduction to VPN

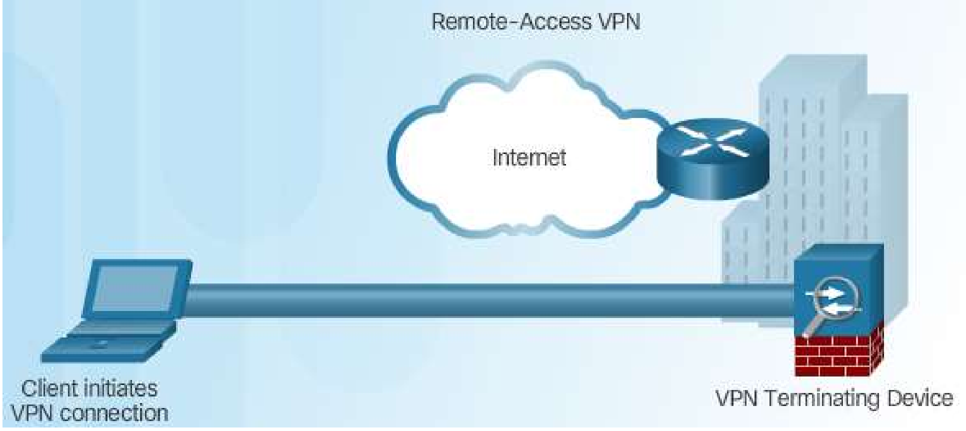
Layer 3 IPsec VPNs



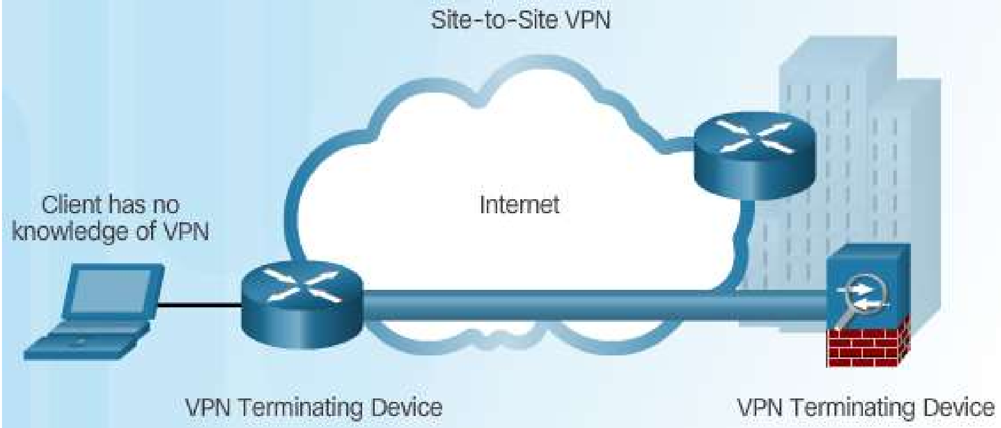
VPN Benefits:

- Cost Savings
- Security
- Scalability
- Compatibility

Types of VPNs

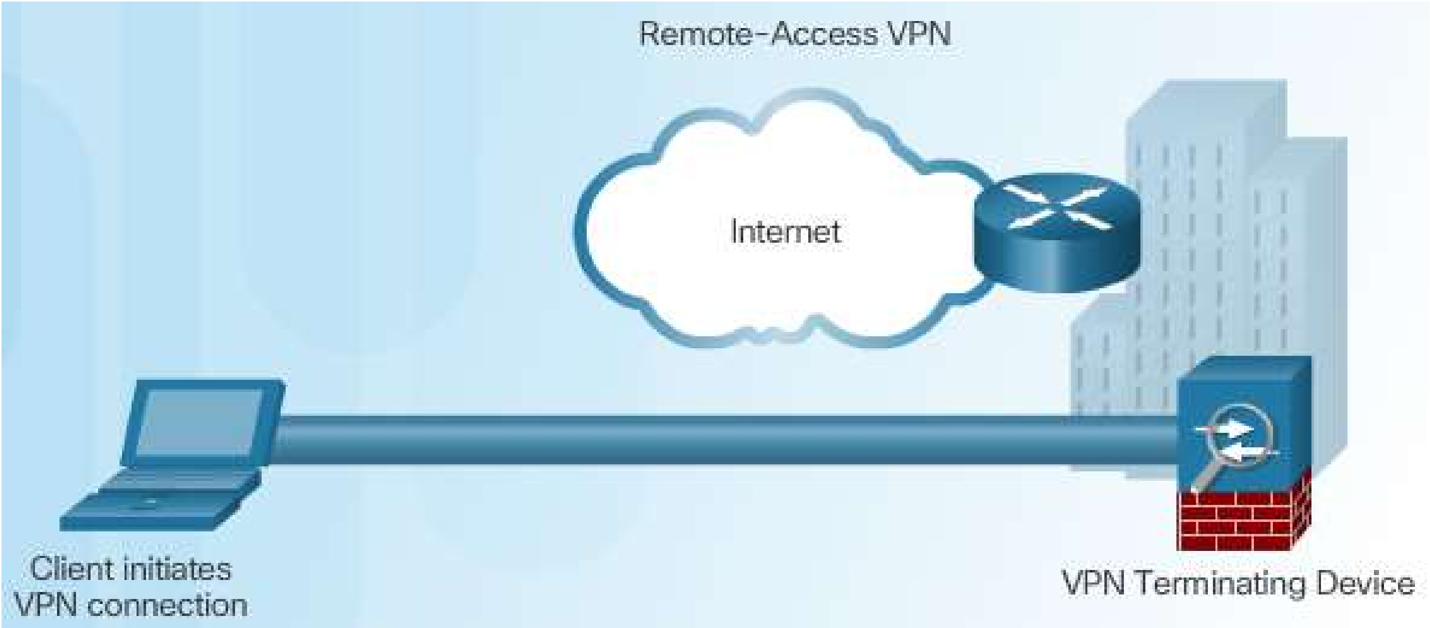


Remote-Access VPN

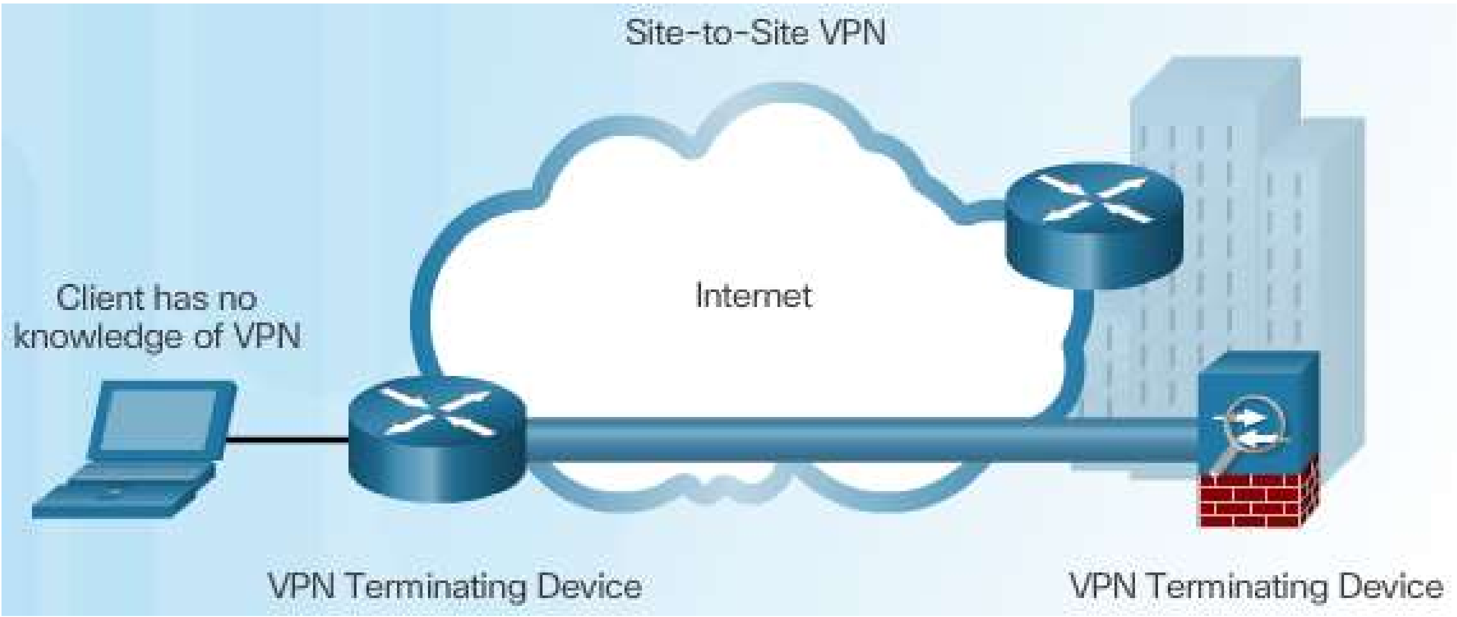


Site-to-Site VPN Access

Components of Remote-Access VPNs

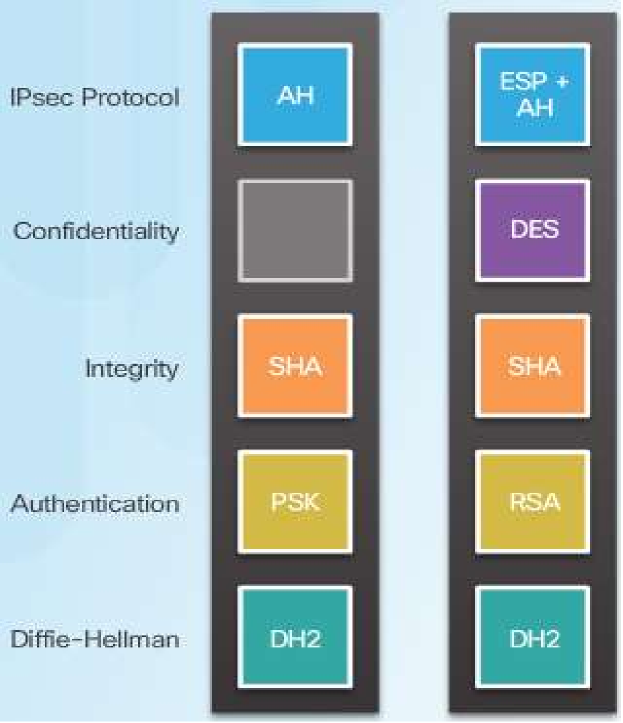


Components of Site-to-Site VPNs



IPsec VPN Components and Operations
IPsec Technologies

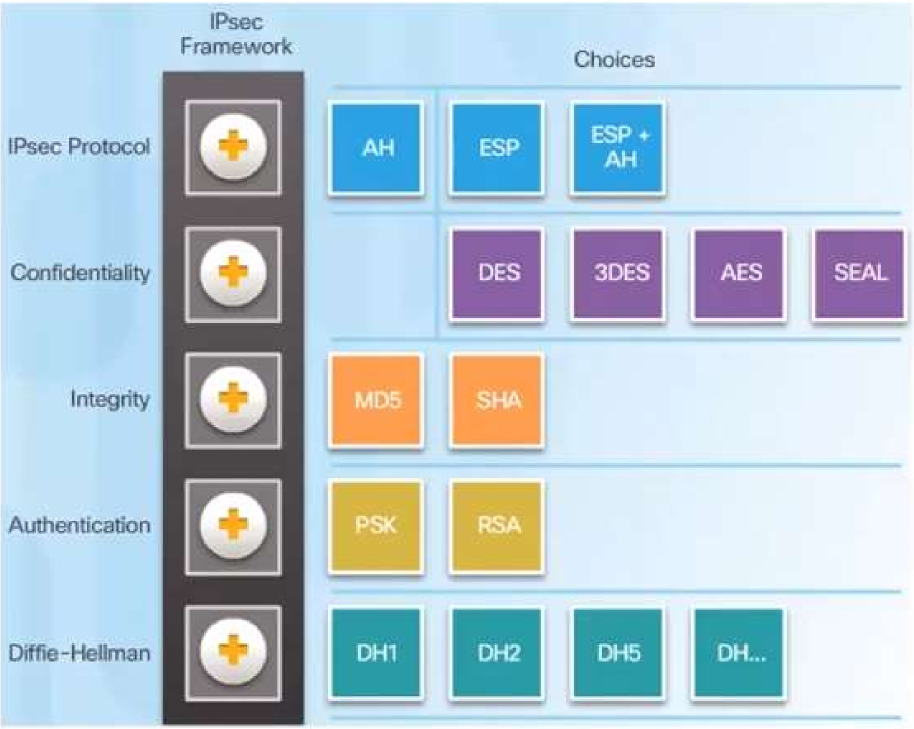
- IPsec Protocols
  - AH (Authentication Header)
  - ESP (Encapsulated Security Payload)
- AH offers integrity and authentication but does not offer any encryption
- ESP not only offers authentication and integrity, but also encrypts the payload



IPsec Protocol	AH	ESP + AH
Confidentiality		DES
Integrity	SHA	SHA
Authentication	PSK	RSA
Diffie-Hellman	DH2	DH2

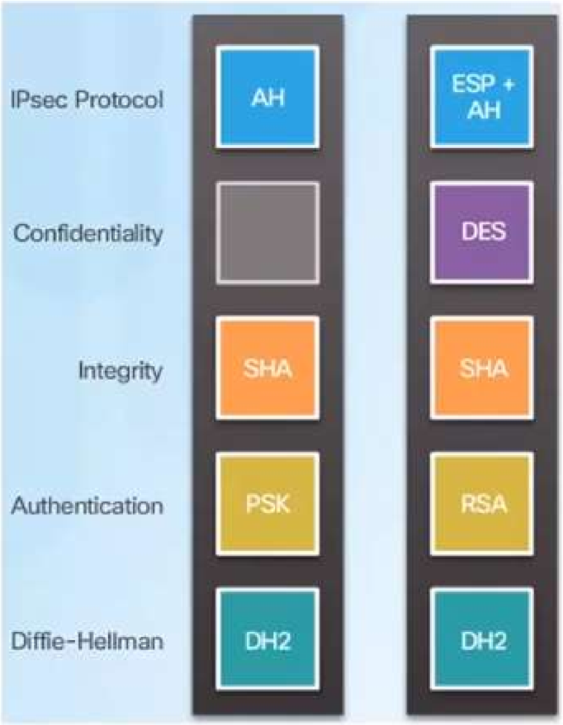
π IPsec Technologies

IPsec Framework



IPsec Framework	Choices			
IPsec Protocol	AH	ESP	ESP + AH	
Confidentiality	DES	3DES	AES	SEAL
Integrity	MD5	SHA		
Authentication	PSK	RSA		
Diffie-Hellman	DH1	DH2	DH5	DH...

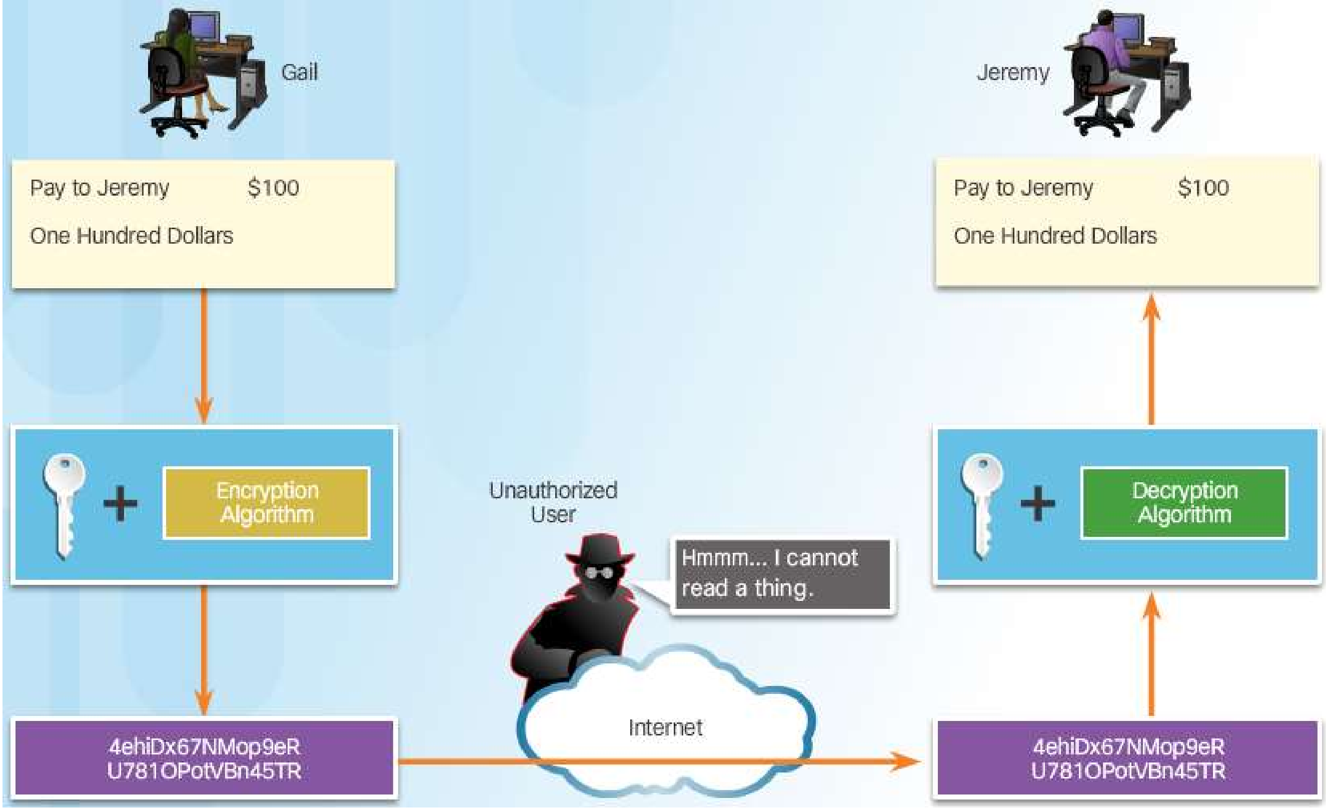
IPsec Implementation Examples



IPsec Protocol	AH	ESP + AH
Confidentiality		DES
Integrity	SHA	SHA
Authentication	PSK	RSA
Diffie-Hellman	DH2	DH2

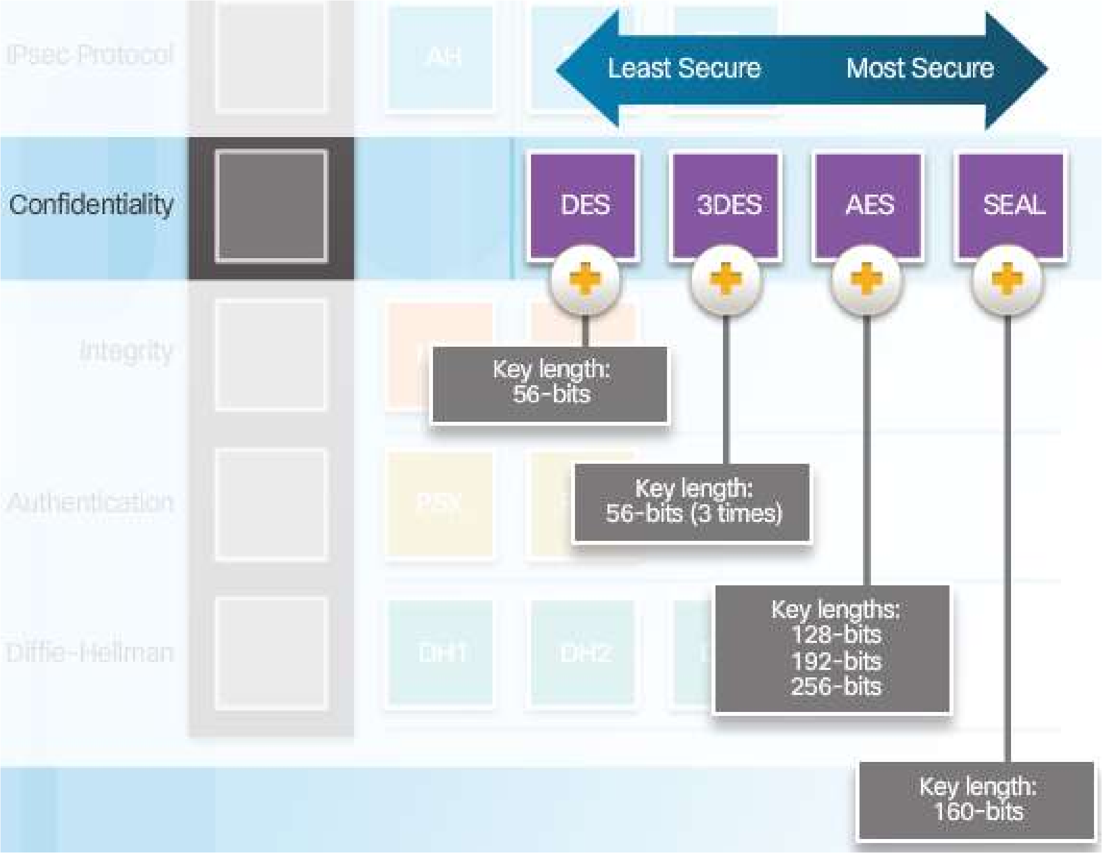
Confidentiality

Confidentiality with Encryption:



Confidentiality

Encryption Algorithms:



Key length: 56-bits

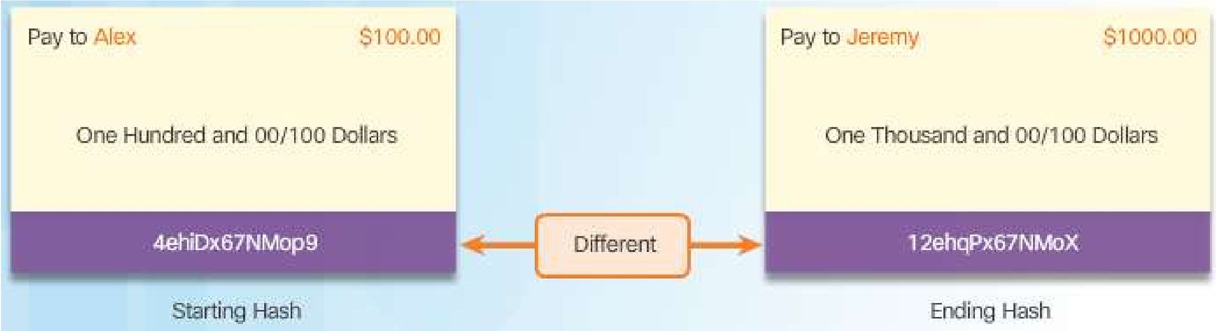
Key length: 56-bits (3 times)

Key lengths: 128-bits 192-bits 256-bits

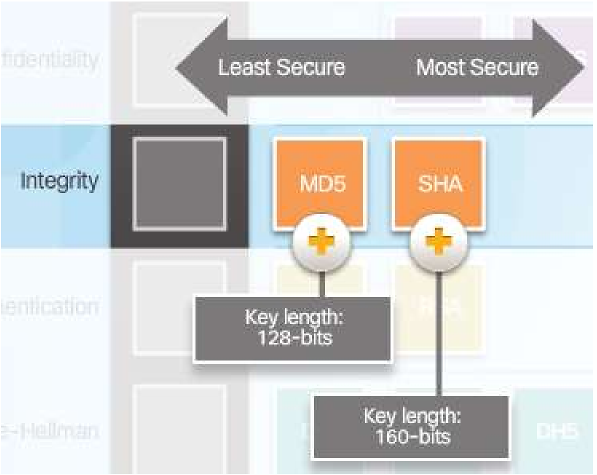
Key length: 160-bits

Integrity

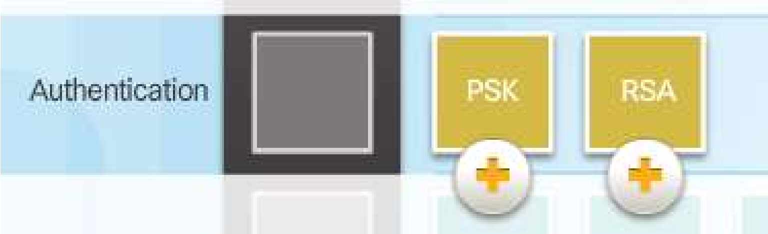
Hash Algorithms



Security of Hash Algorithms

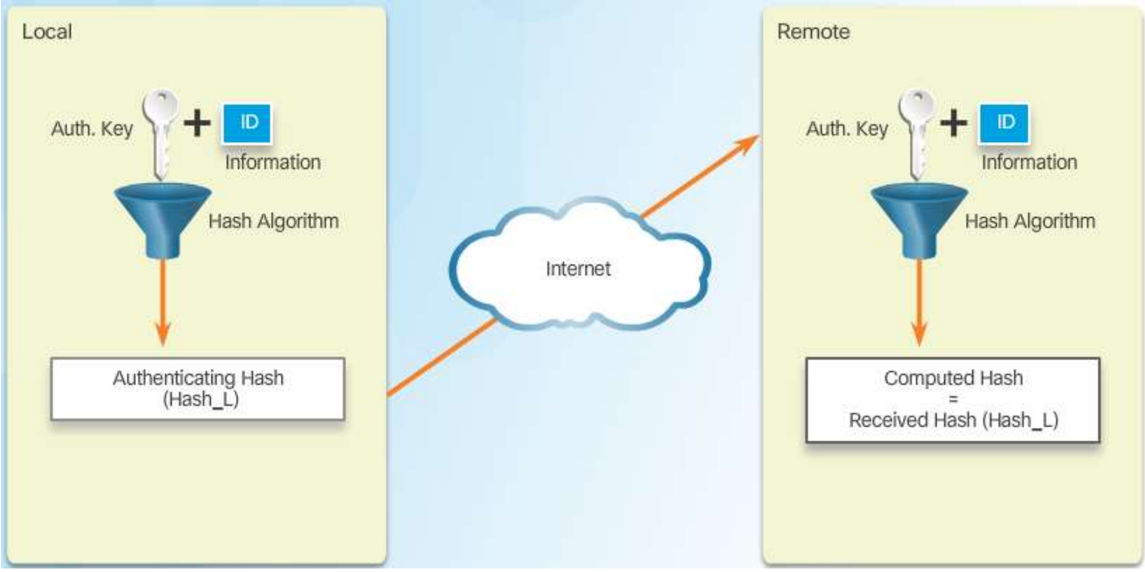


Authentication



Peer Authentication Methods

PSK (Pre-shared Key)

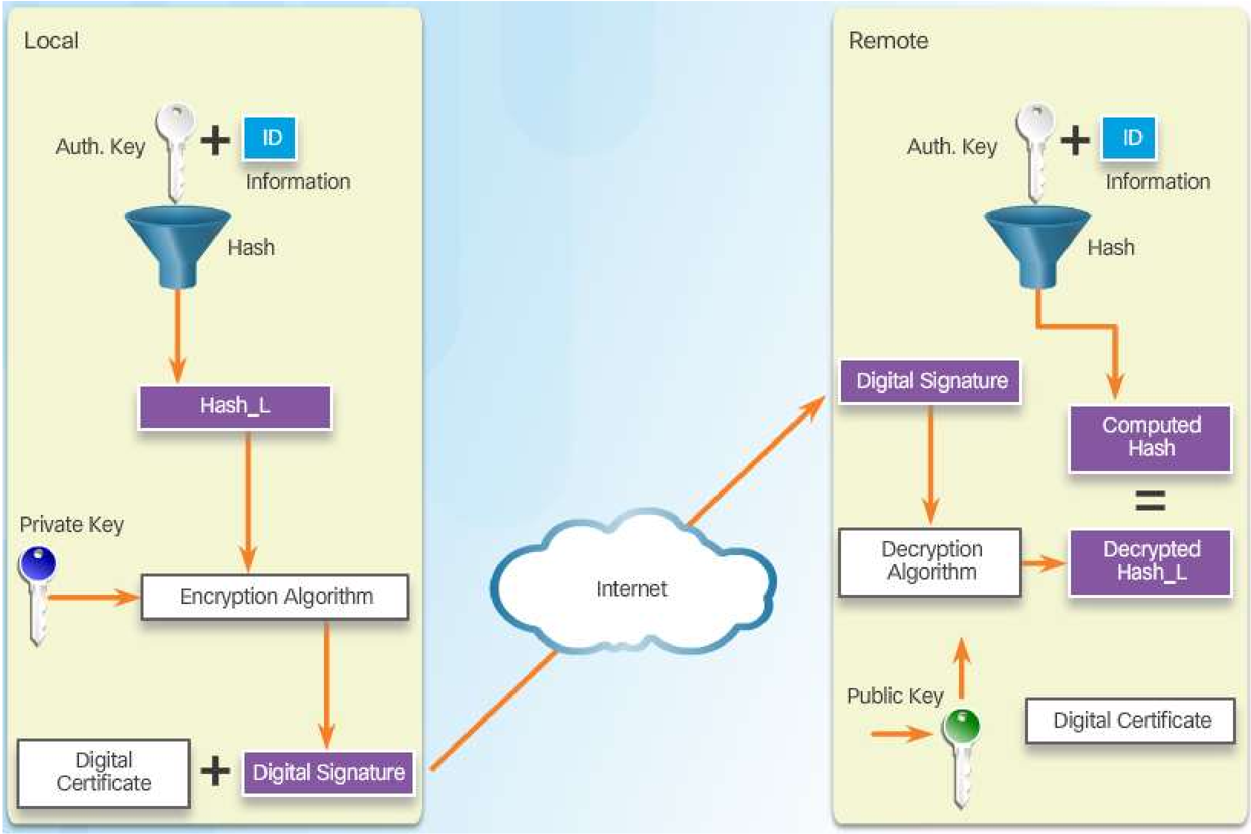


Authentication - PSK

- At the local device, the authentication key and the device identity are sent through a hash algorithm to form hash_L (Authenticating Hash). One-way authentication is established by sending hash_L to the remote device. If the remote device computes the same hash, then the local device is authenticated
- For the authentication in the opposite direction, the remote device combines its identity with the preshared-based key and send it through the hash algorithm to form hash_R, then hash_R is sent to the local device. If the local device can compute the same hash, the remote device is authenticated.

Authentication

RSA (Rivest-Shamir-Adleman)

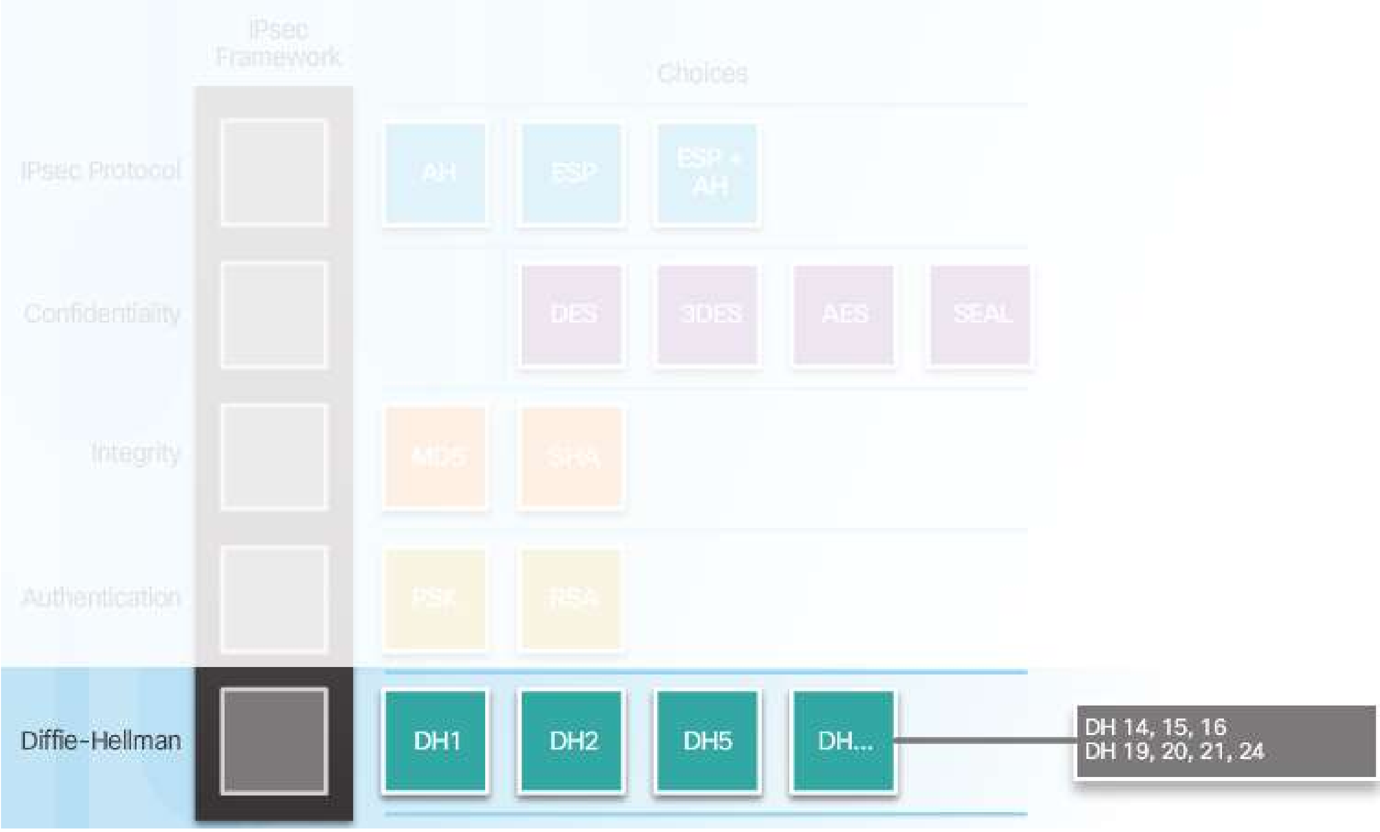


Authentication - RSA

- At the local device, the authentication key and the device identity are sent through a hash algorithm to form hash_L. hash_L is encrypted using the local device's private encryption key creating a digital signature. The public encryption key for decrypting the signature is included in the digital certificate.
- The remote device verifies the digital signature by decrypting it using the public encryption key. The result is hash_L.
- Next, the remote device creates hash_L from stored information. If the computed hash_L = decrypted hash_L, then the local device is authenticated. After that, the authentication process in the opposite direction begins by repeating the same steps.

Secure Key Exchange

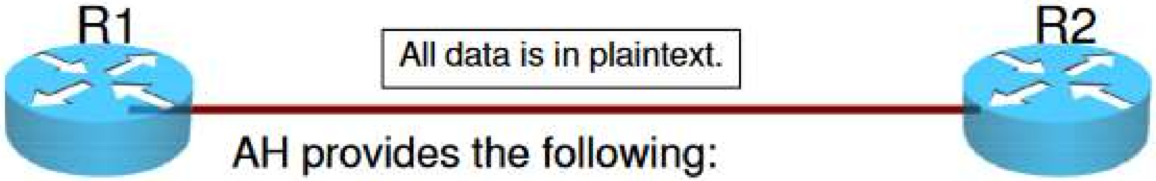
Diffie-Hellman Key Exchange



DH 14, 15, 16
DH 19, 20, 21, 24

IPsec Protocols

AH Protocols

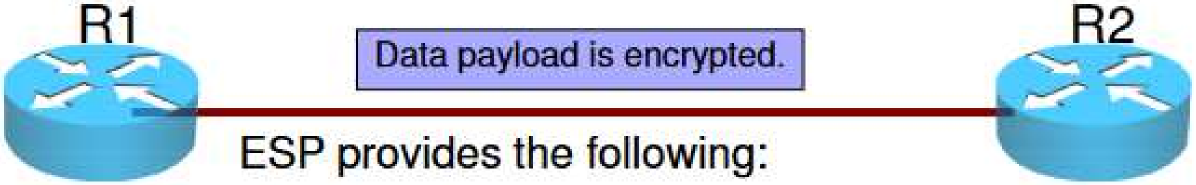


All data is in plaintext.

AH provides the following:

- Authentication
- Integrity

ESP Protocols

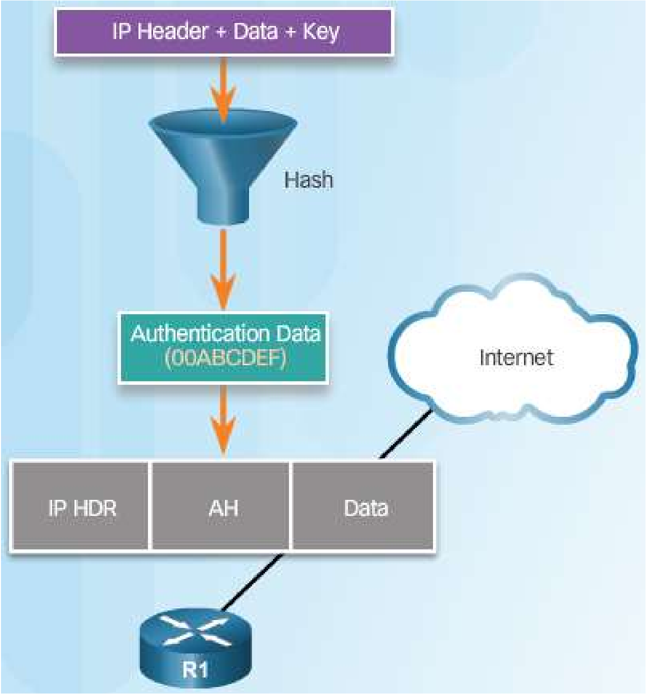


Data payload is encrypted.

ESP provides the following:

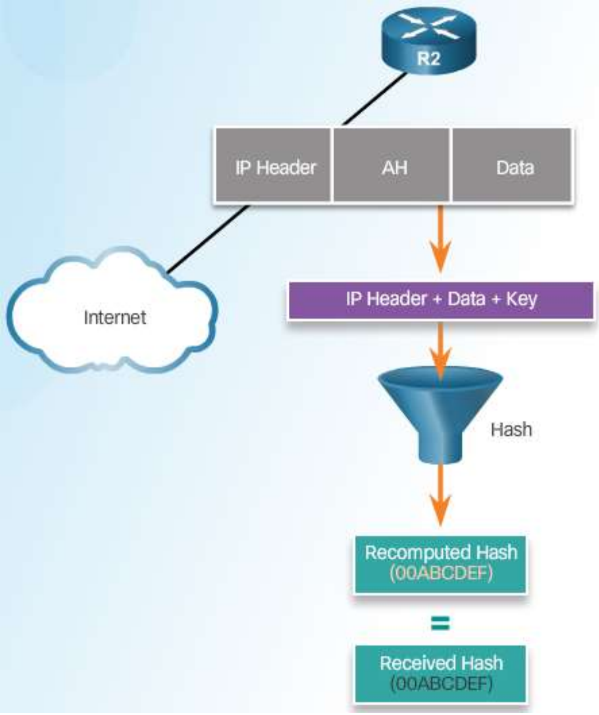
- **Encryption**
- Authentication
- Integrity

Authentication Header

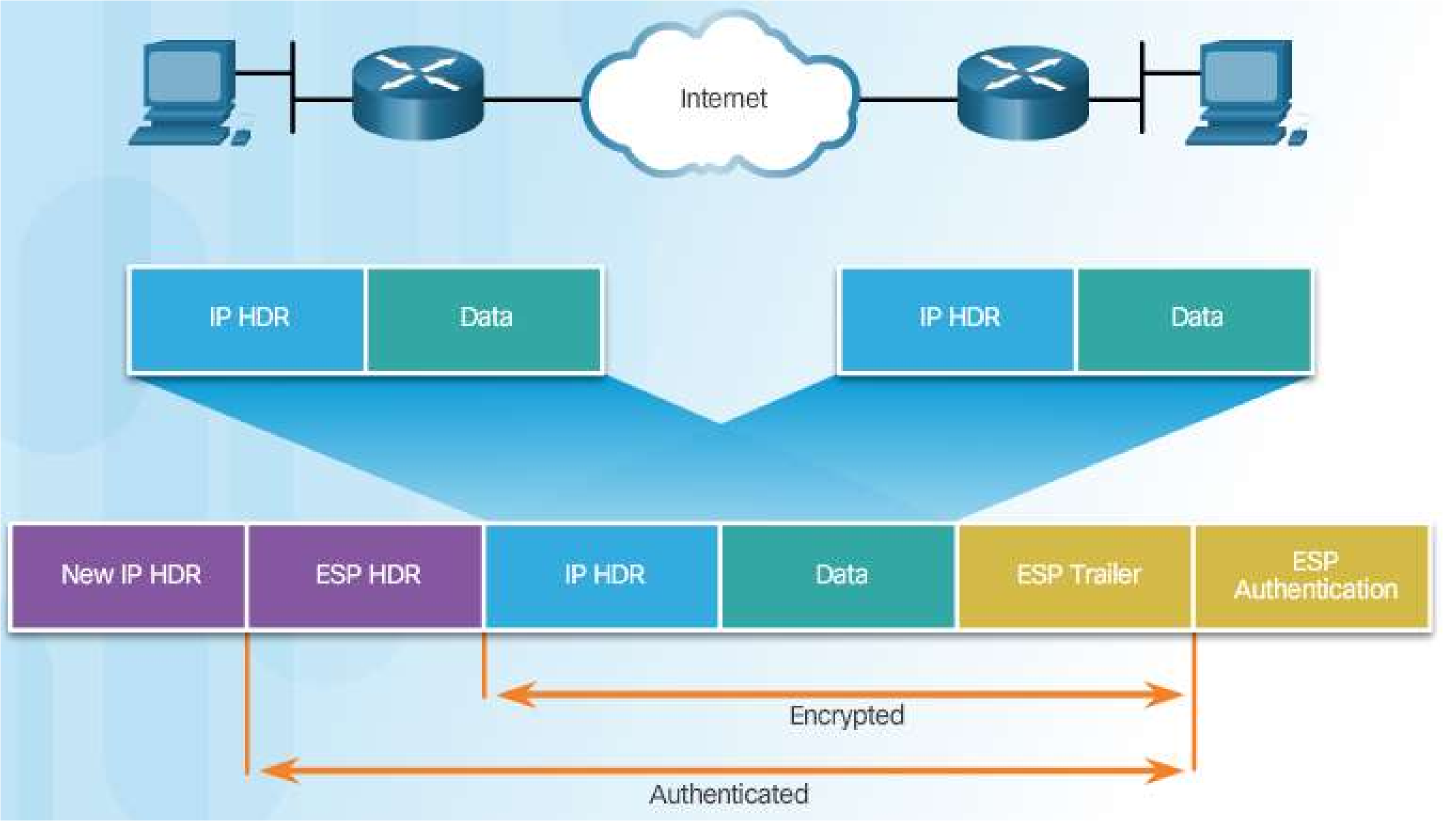


Router Creates Hash and Transmits to Peer

Peer Router hashes the IP header and data payload, then compares this Recomputed Hash to Received Hash

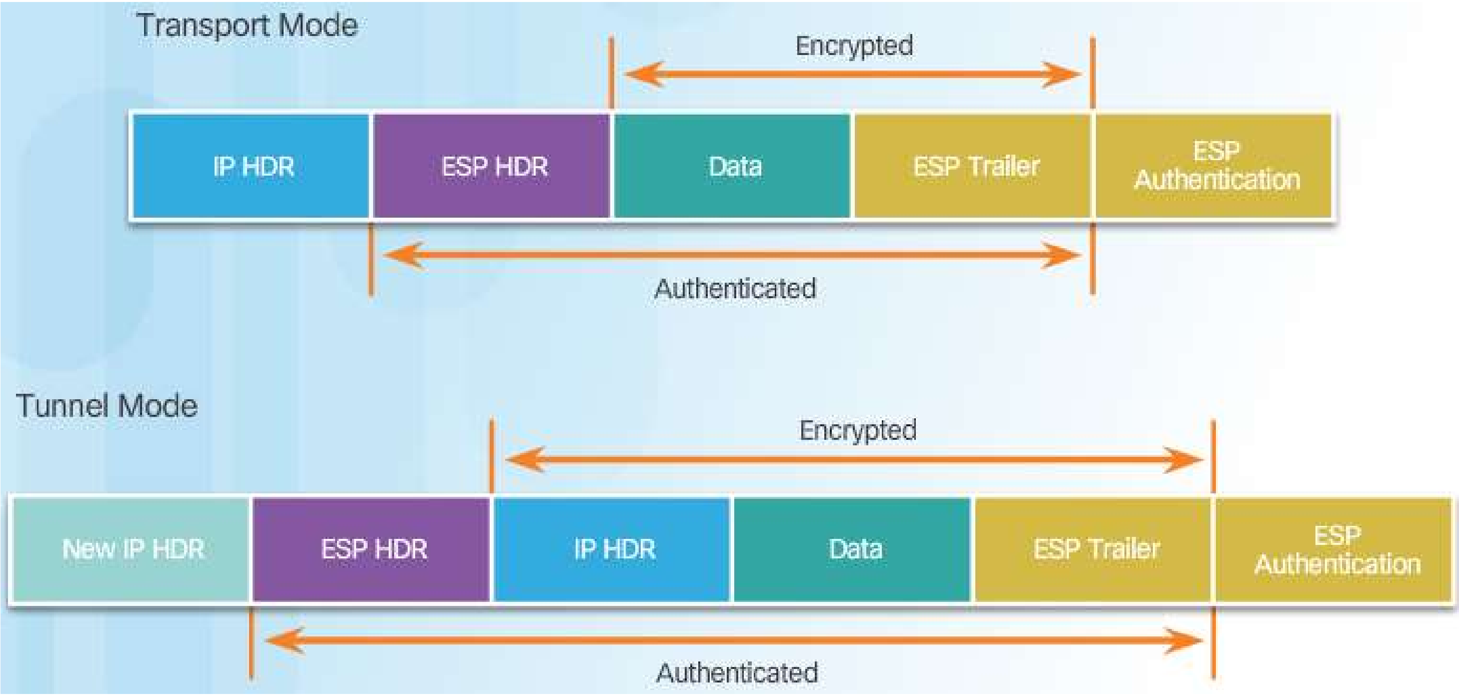


ESP Encrypts and Authenticates



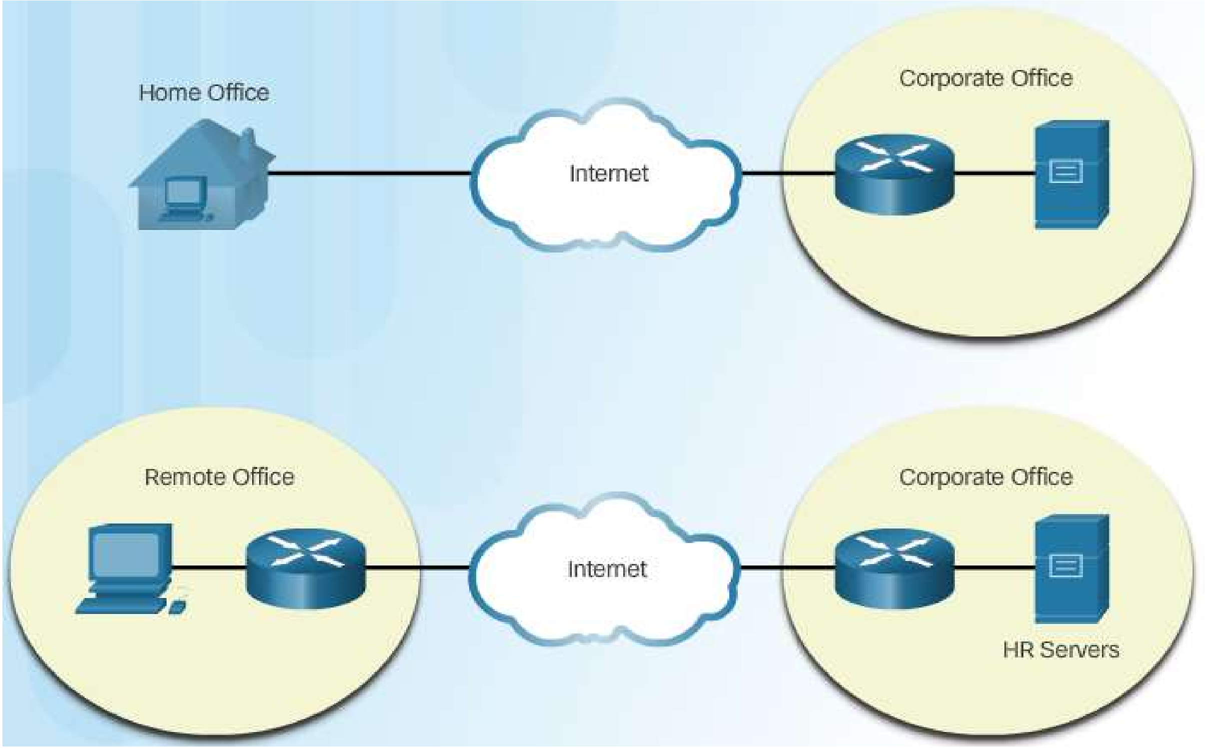
Transport and Tunnel Modes

ESP in Two Modes



Transport and Tunnel Modes (Cont.)

ESP in Tunnel Mode



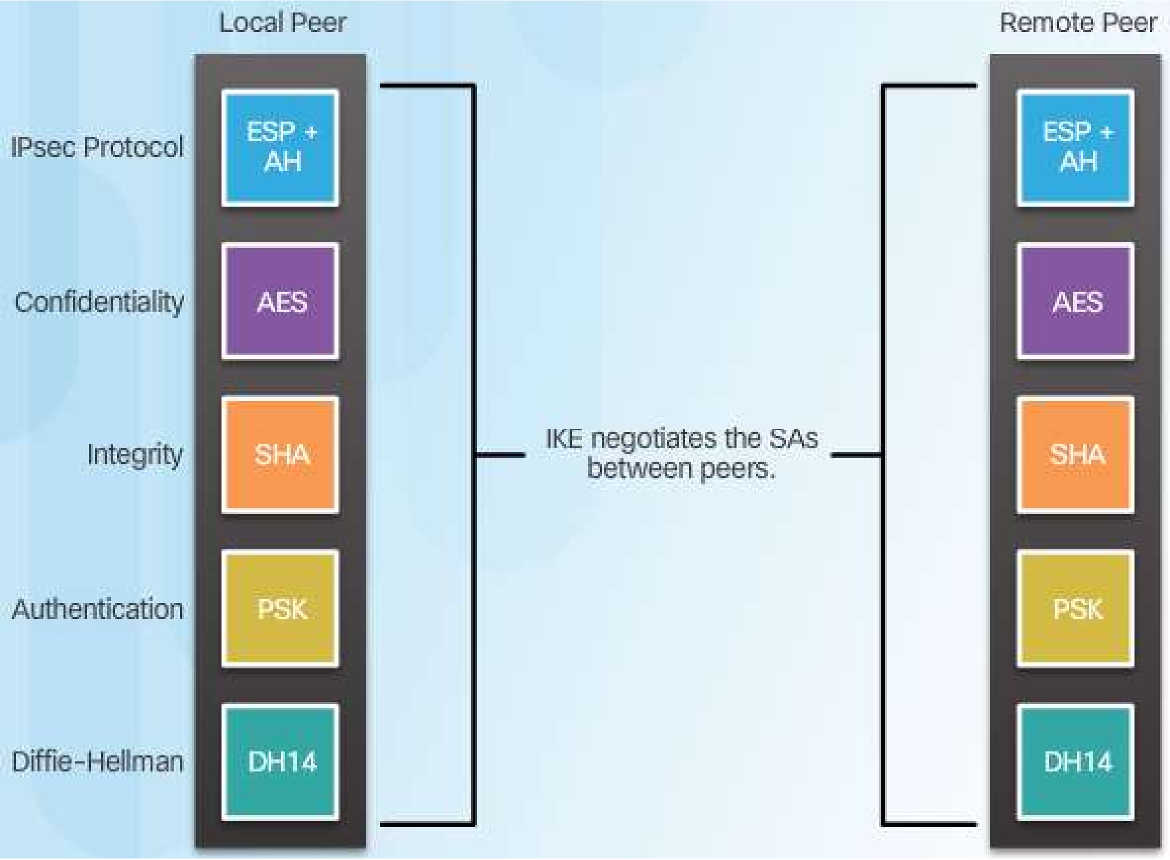
Transport Mode

- Provides protection primarily for upper-layer protocols
  - Examples include a TCP or UDP segment or an ICMP packet
- Typically used for end-to-end communication between two hosts
- ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header
- AH in transport mode authenticates the IP payload and selected portions of the IP header

Tunnel Mode

- Provides protection to the entire IP packet
- Used when one or both ends of a security association (SA) are a security gateway
- A number of hosts on networks behind firewalls may engage in secure communications without implementing IPsec
- ESP in tunnel mode encrypts and optionally authenticates the entire inner IP packet, including the inner IP header
- AH in tunnel mode authenticates the entire inner IP packet and selected portions of the outer IP header

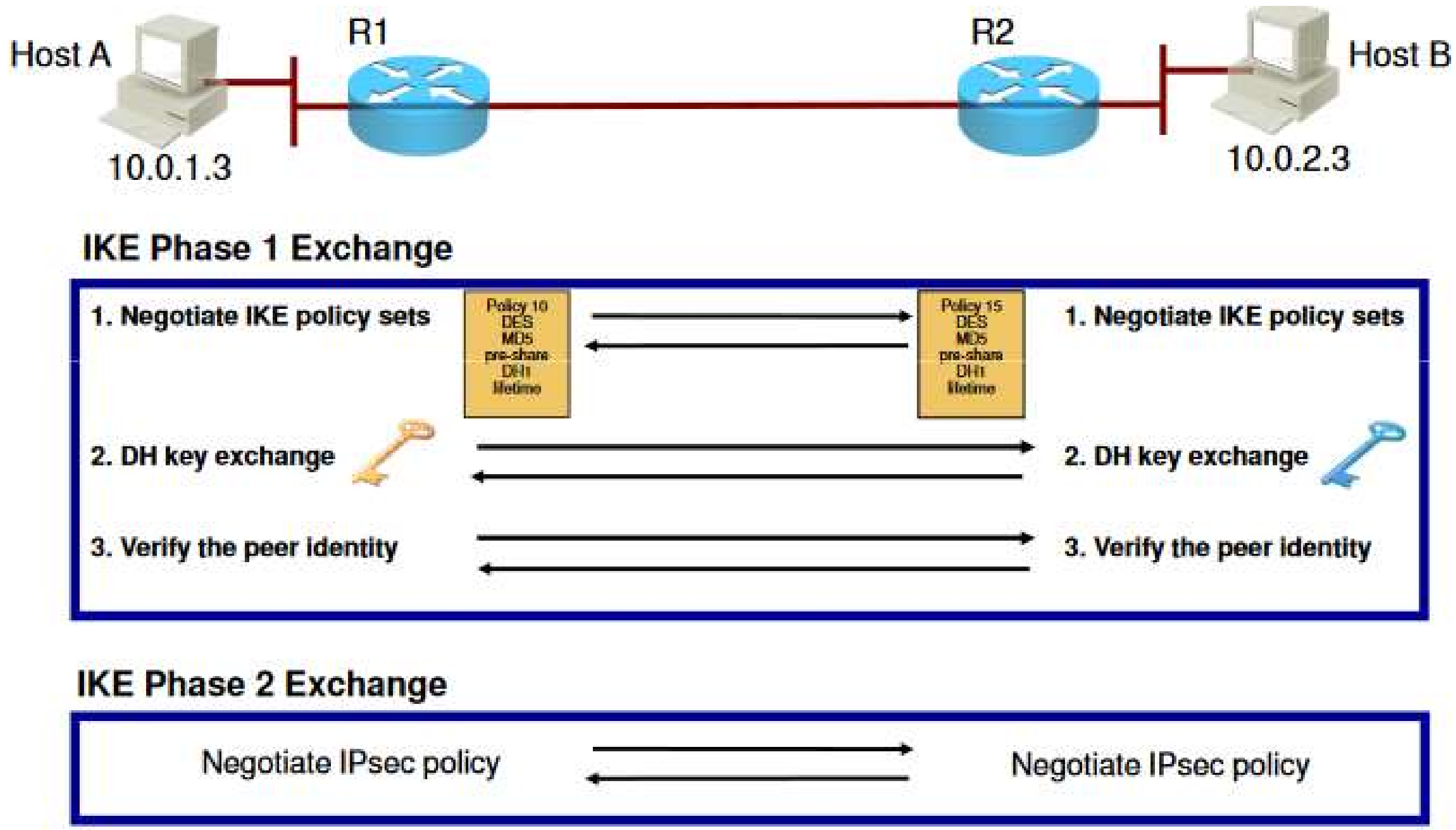
The Internet Key Exchange (IKE) Protocol



	Local Peer	Remote Peer
IPsec Protocol	ESP + AH	ESP + AH
Confidentiality	AES	AES
Integrity	SHA	SHA
Authentication	PSK	PSK
Diffie-Hellman	DH14	DH14

IKE negotiates the SAs between peers.

IKE Phase 1 and 2



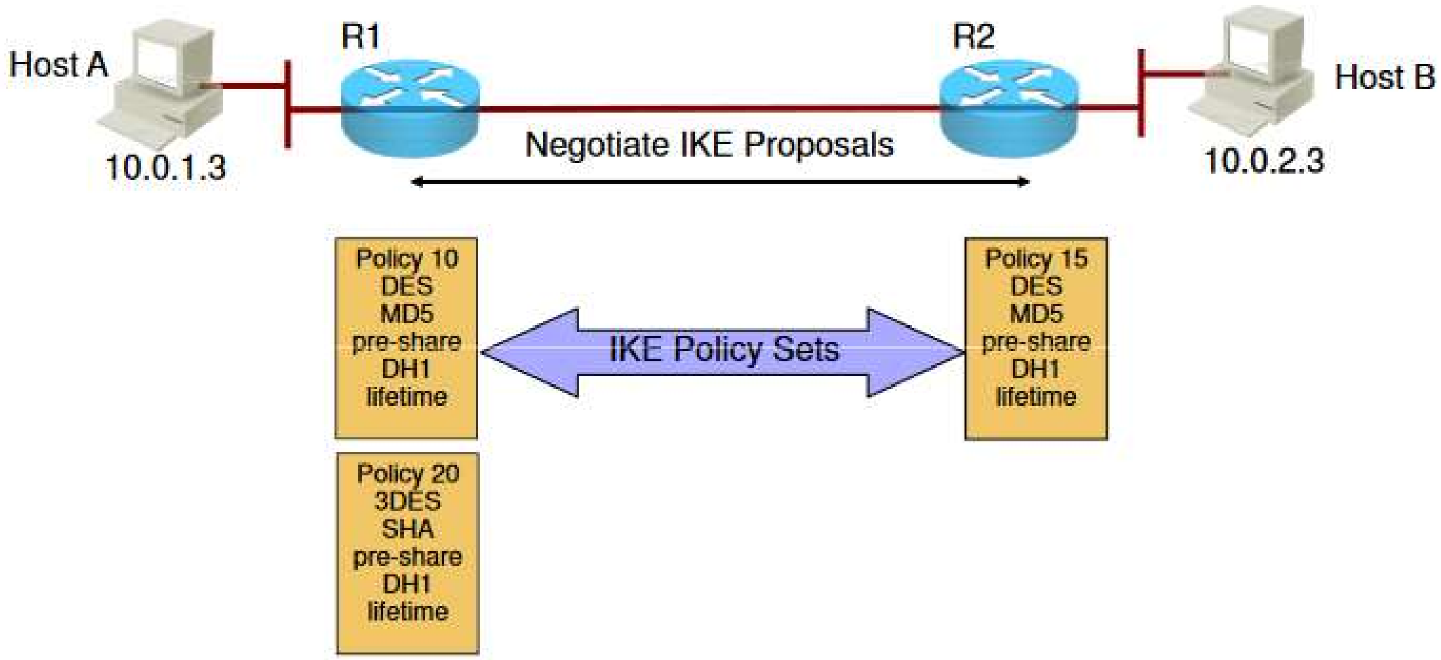
IKE Phase 1 Exchange

1. Negotiate IKE policy sets	**1. Negotiate IKE policy sets**
2. DH key exchange	**2. DH key exchange**
3. Verify the peer identity	**3. Verify the peer identity**

IKE Phase 2 Exchange

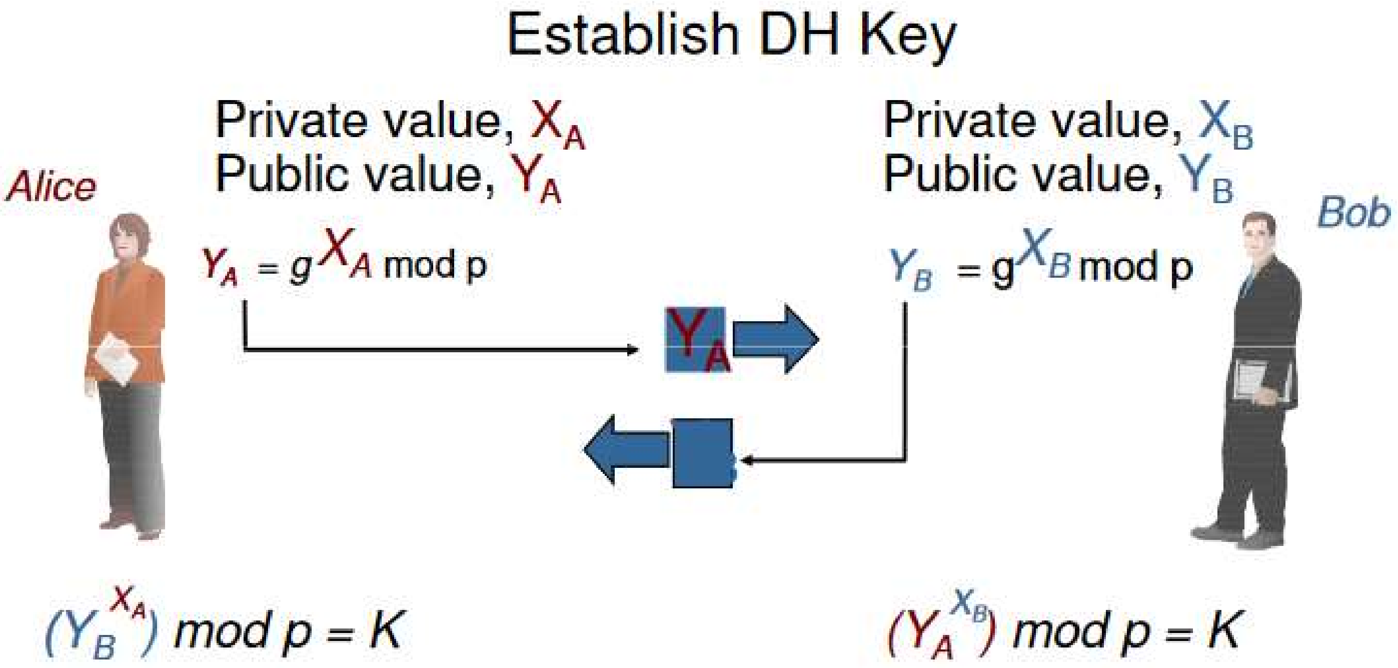
Negotiate IPsec policy	Negotiate IPsec policy

IKE Phase 1 – 1st Exchange



Negotiates matching IKE policies to protect IKE exchange

IKE Phase 1 – 2nd Exchange



Establish DH Key

Alice

Private value, X_A
Public value, Y_A

$Y_A = g^{X_A} \bmod p$

Private value, X_B
Public value, Y_B

Bob

$Y_B = g^{X_B} \bmod p$

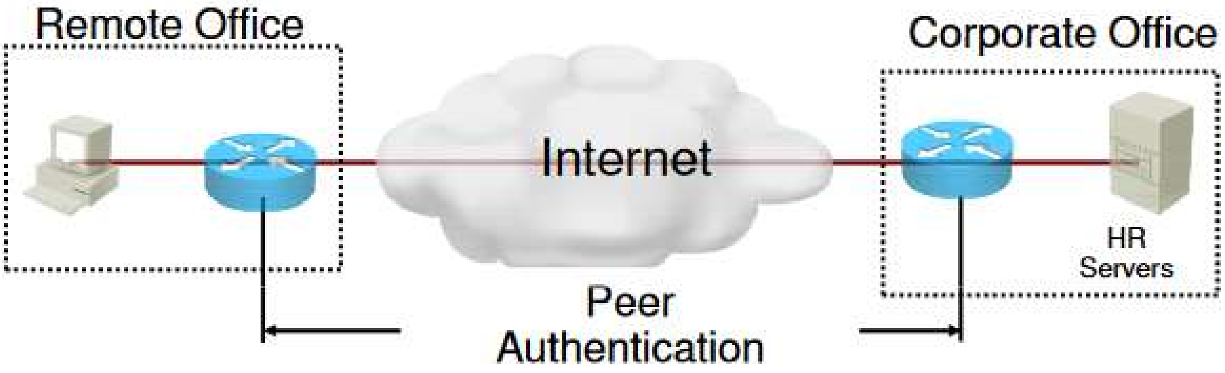
$(Y_B^{X_A}) \bmod p = K$

$(Y_A^{X_B}) \bmod p = K$

A DH exchange is performed to establish keying material.

IKE Phase 1 – 3rd Exchange

Authenticate Peer

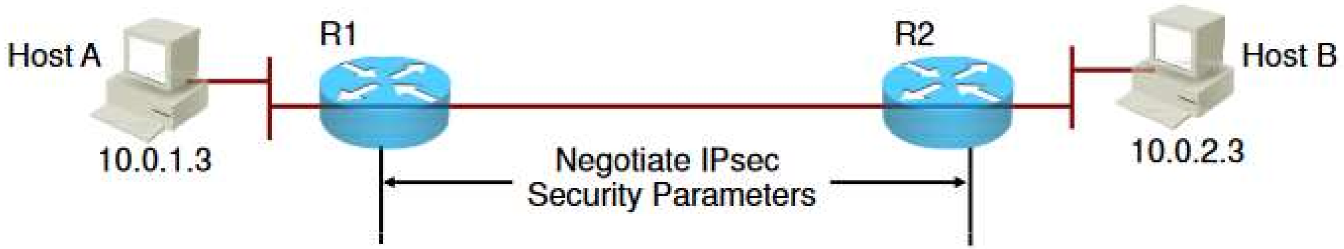


Peer authentication methods

- PSKs
- RSA signatures
- RSA encrypted nonces

A bidirectional IKE SA is now established.

IKE Phase 2: Negotiating SAs

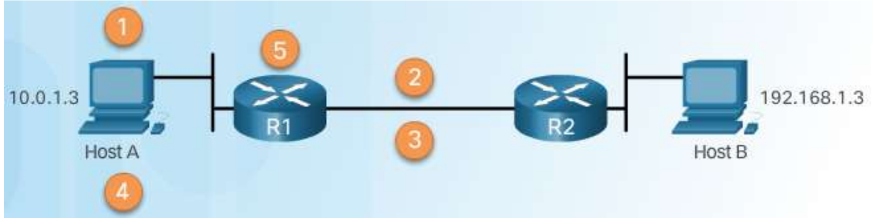


- IKE negotiates matching IPsec policies
- Upon completion, IPsec Security Associations (SAs) are established for each protocol and algorithm combination

Implementing Site-to-Site IPsec VPNs

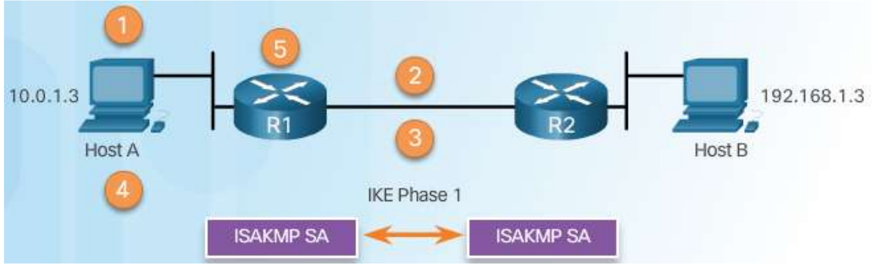
- IPsec negotiation and the five steps of IPsec configuration.
- Configure compatible ACLs.
- Configure the ISAKMP policy.
- Configure IPsec transform set
- Create a crypto ACL
- Configure and apply a crypto map.

IPsec Negotiation & 5-steps of IPsec Configuration

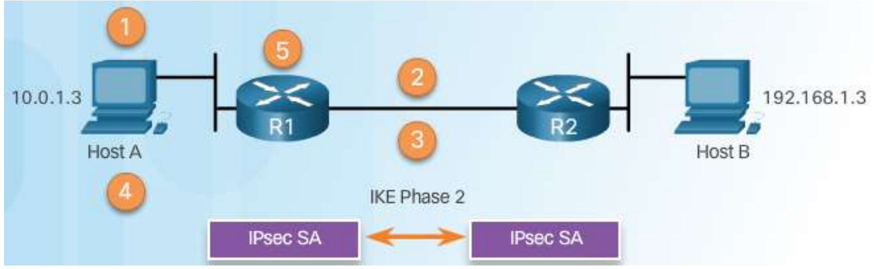


Step 1 - Host A sends interesting traffic to Host B.

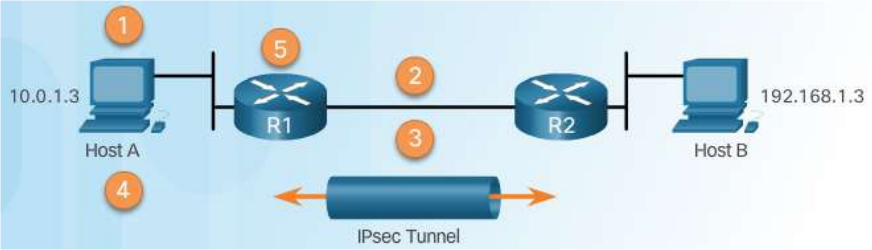
Step 2 - R1 and R2 negotiate an IKE Phase 1 session.



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

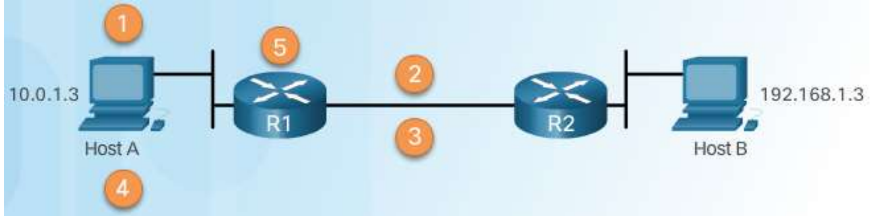


IPsec Negotiation & 5-steps of IPsec Configuration

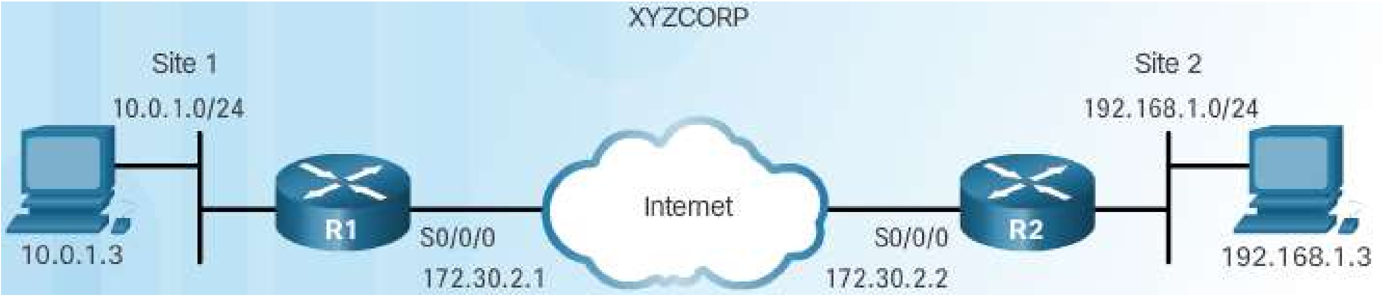


Information is exchanged via IPsec tunnel.

Step 5 - The IPsec tunnel is terminated.

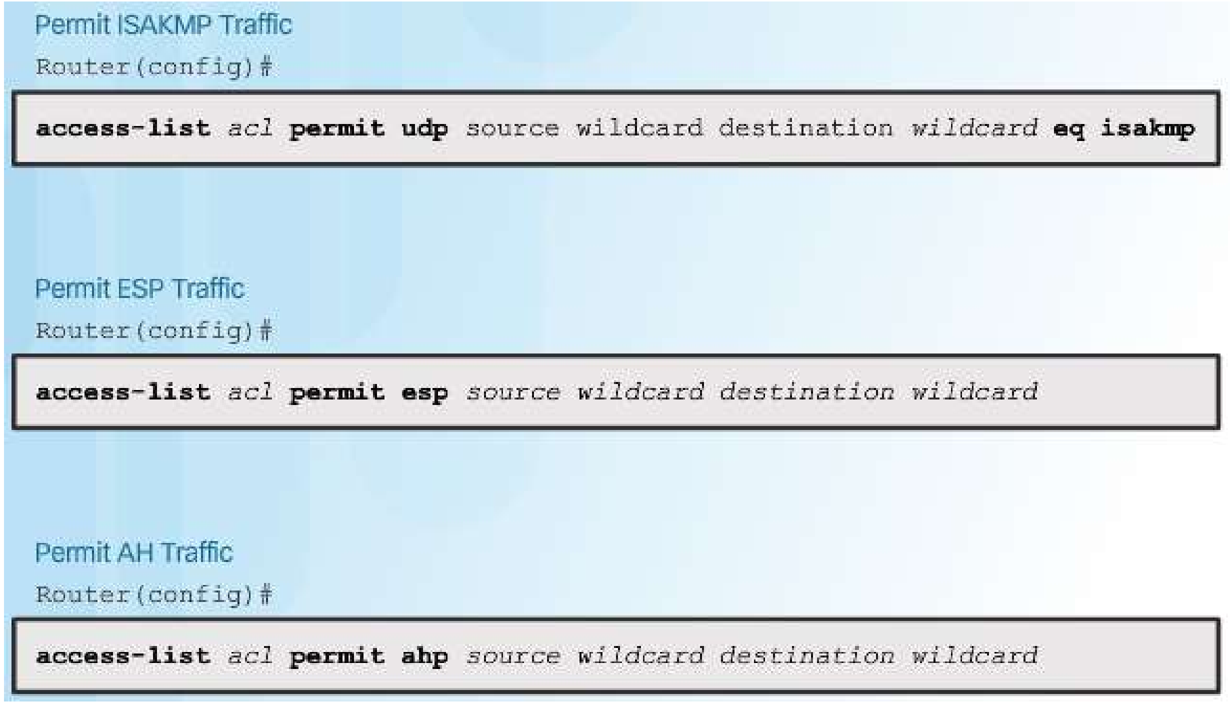


IPsec VPN Configuration Tasks



XYZCORP Security Policy	Configuration Tasks
Encrypt traffic with AES 256 and SHA	1. Configure compatible ACLs
Authentication with PSK	2. Configure the ISAKMP policy
Exchange keys with group 24	3. Configure IPsec transform set
ISAKMP tunnel lifetime is 1 hour	4. Create a crypto ACL
IPsec tunnel uses ESP with a 15-min. lifetime	5. Configure and apply a crypto map

Configure Compatible ACLs



Permit ISAKMP Traffic

```
Router(config)#
access-list acl permit udp source wildcard destination wildcard eq isakmp
```

Permit ESP Traffic

```
Router(config)#
access-list acl permit esp source wildcard destination wildcard
```

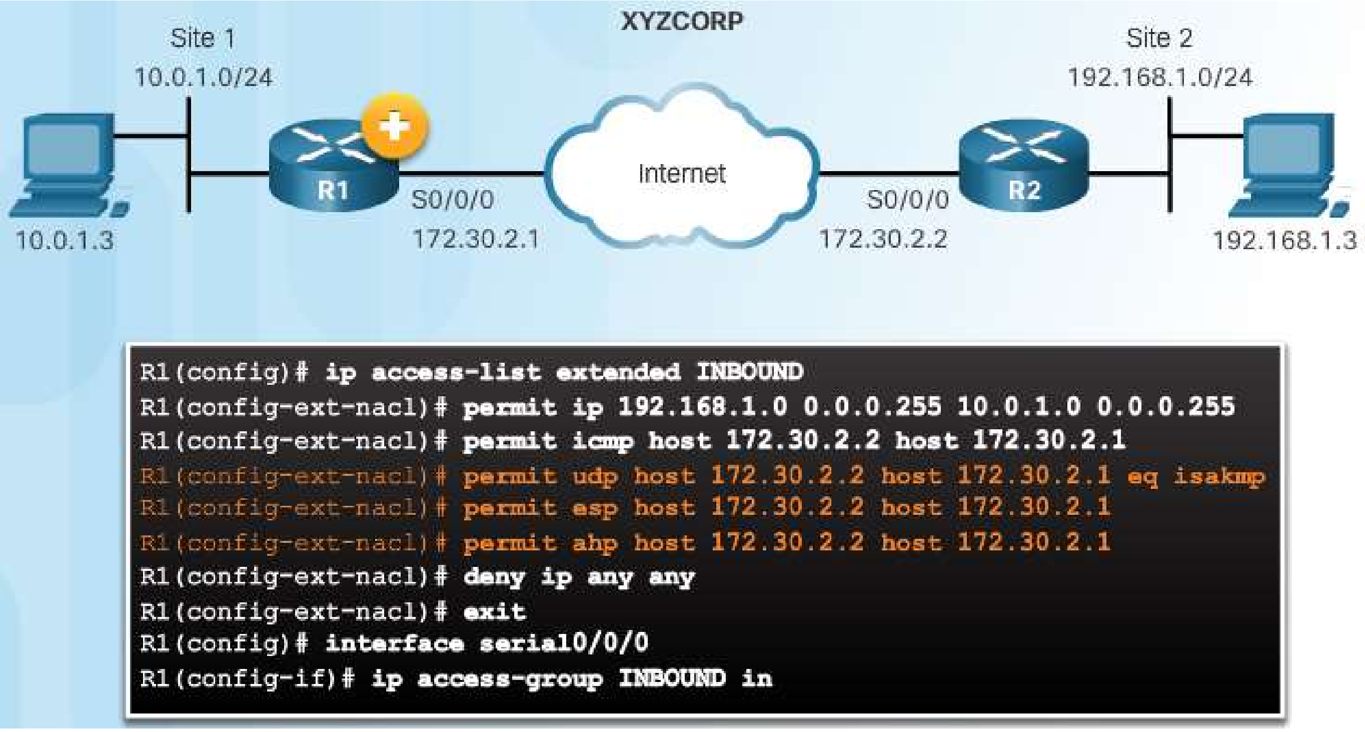
Permit AH Traffic

```
Router(config)#
access-list acl permit ahp source wildcard destination wildcard
```

ACL Syntax for IPsec Traffic

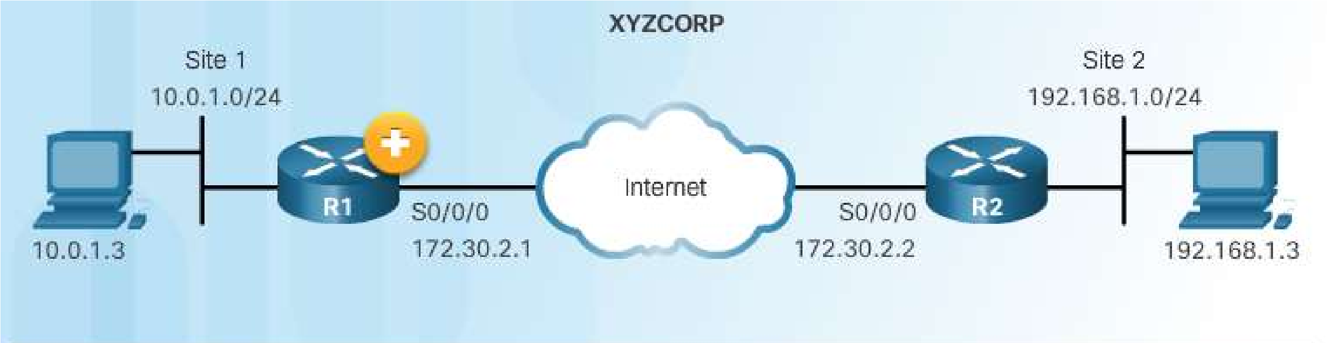
Configure Compatible ACLs

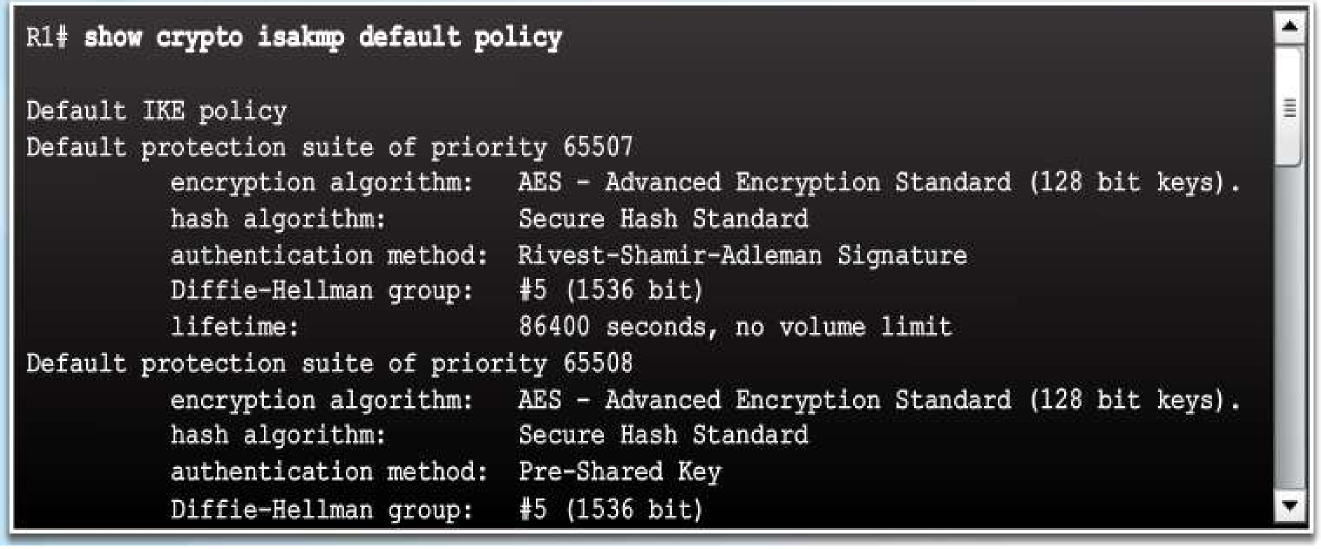
Permitting Traffic for IPsec Negotiations



```
R1(config)# ip access-list extended INBOUND
R1(config-ext-nacl)# permit ip 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
R1(config-ext-nacl)# permit icmp host 172.30.2.2 host 172.30.2.1
R1(config-ext-nacl)# permit udp host 172.30.2.2 host 172.30.2.1 eq isakmp
R1(config-ext-nacl)# permit esp host 172.30.2.2 host 172.30.2.1
R1(config-ext-nacl)# permit ahp host 172.30.2.2 host 172.30.2.1
R1(config-ext-nacl)# deny ip any any
R1(config-ext-nacl)# exit
R1(config)# interface serial0/0/0
R1(config-if)# ip access-group INBOUND in
```

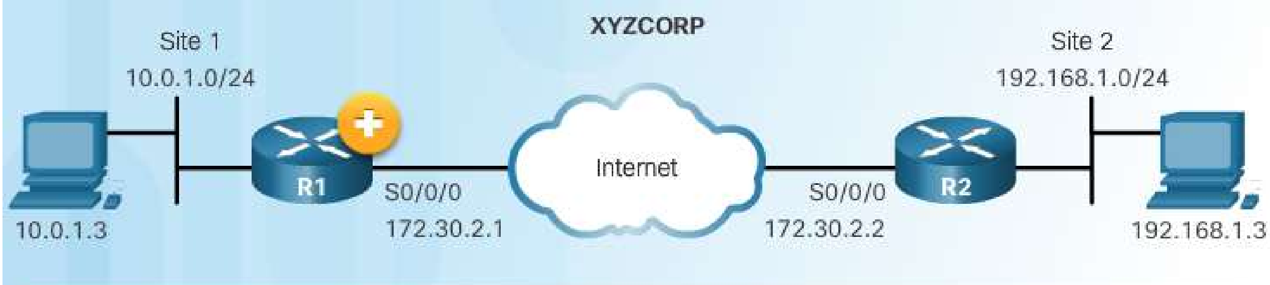
The Default ISAKMP Policies

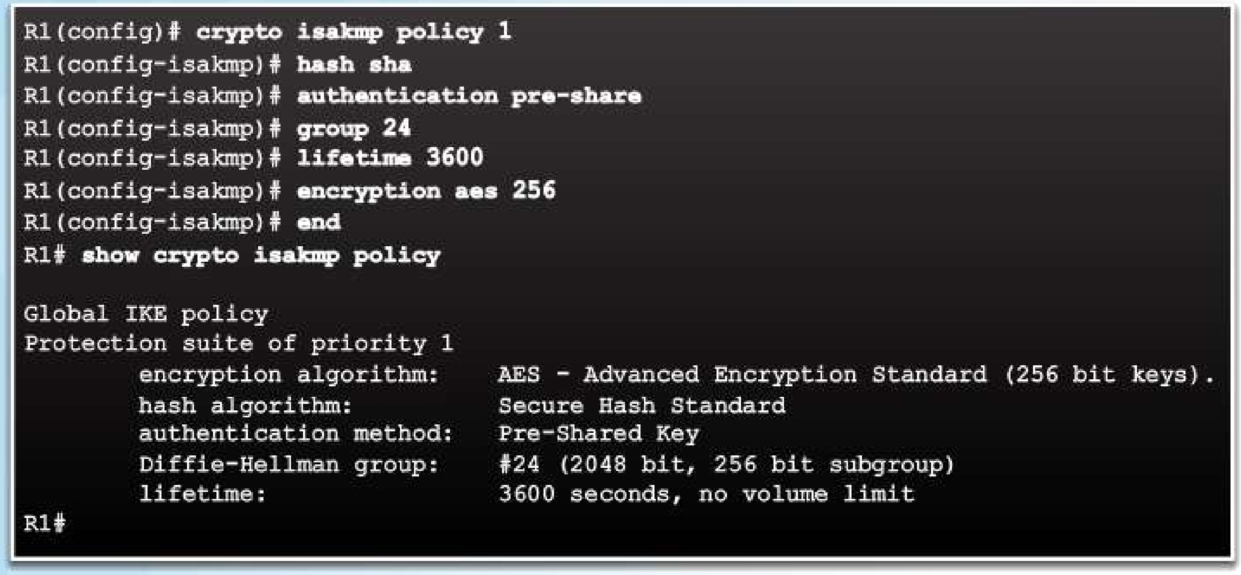




```
R1# show crypto isakmp default policy

Default IKE policy
Default protection suite of priority 65507
          encryption algorithm:   AES - Advanced Encryption Standard (128 bit keys).
          hash algorithm:         Secure Hash Standard
          authentication method:  Rivest-Shamir-Adleman Signature
          Diffie-Hellman group:   #5 (1536 bit)
          lifetime:               86400 seconds, no volume limit
Default protection suite of priority 65508
          encryption algorithm:   AES - Advanced Encryption Standard (128 bit keys).
          hash algorithm:         Secure Hash Standard
          authentication method:  Pre-Shared Key
          Diffie-Hellman group:   #5 (1536 bit)
```

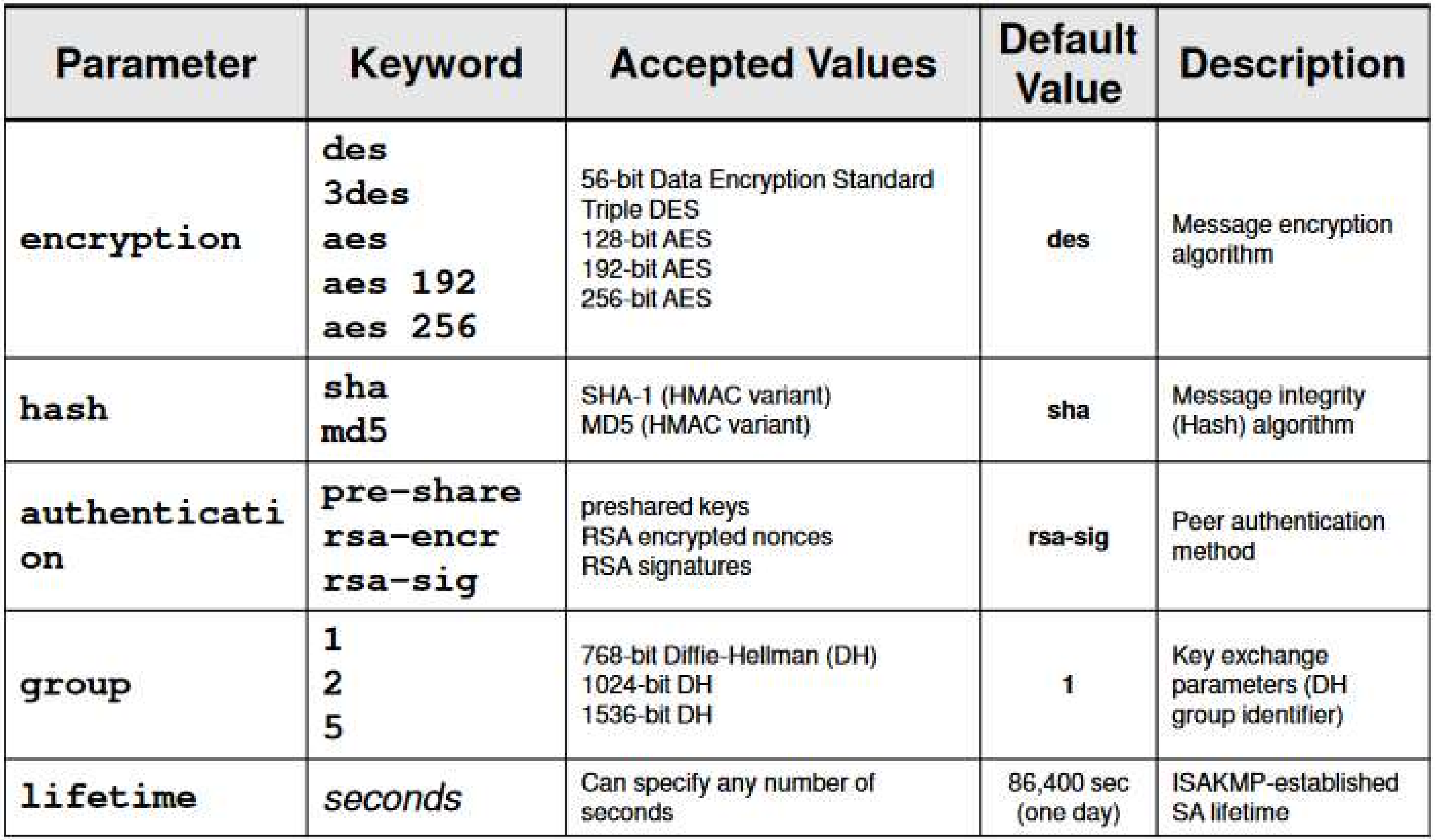
Configure a New ISAKMP Policy





```
R1(config)# crypto isakmp policy 1
R1(config-isakmp)# hash sha
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 24
R1(config-isakmp)# lifetime 3600
R1(config-isakmp)# encryption aes 256
R1(config-isakmp)# end
R1# show crypto isakmp policy

Global IKE policy
Protection suite of priority 1
        encryption algorithm:    AES - Advanced Encryption Standard (256 bit keys).
        hash algorithm:          Secure Hash Standard
        authentication method:   Pre-Shared Key
        Diffie-Hellman group:    #24 (2048 bit, 256 bit subgroup)
        lifetime:                3600 seconds, no volume limit
R1#
```

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

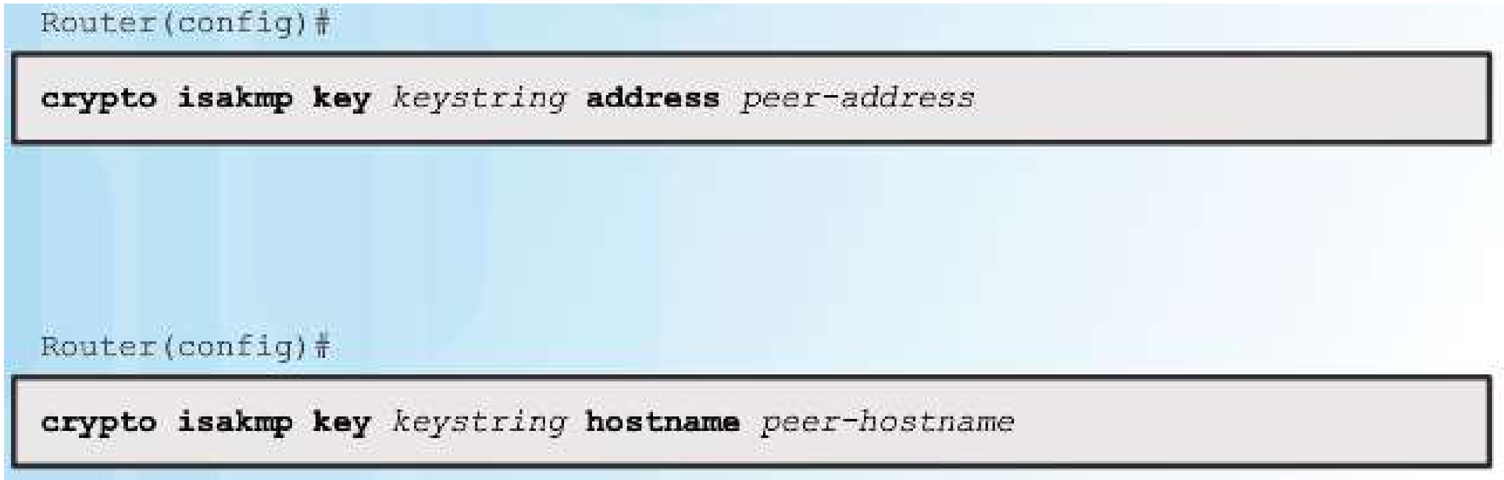
`authentication`	`pre-share`
	`rsa-encr`
	`rsa-sig`
		preshared keys
		RSA encrypted nonces
		RSA signatures
			rsa-sig	Peer authentication method

`group`	`1`
	`2`
	`5`
		768-bit Diffie-Hellman (DH)
		1024-bit DH
		1536-bit DH
			1	Key exchange parameters (DH group identifier)

`lifetime`	*seconds*	Can specify any number of seconds	86,400 sec (one day)	ISAKMP-established SA lifetime

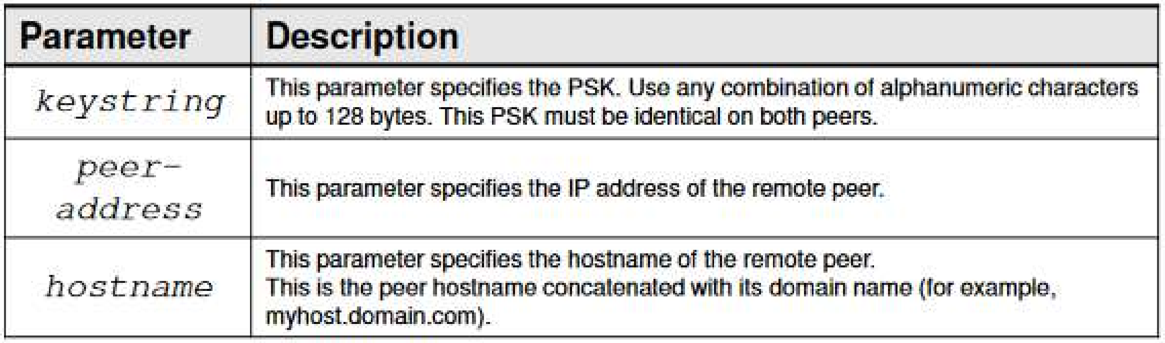
Configuring a Pre-Shared Key

The `crypto isakmp key` Command



```
Router(config)#
crypto isakmp key keystring address peer-address
```

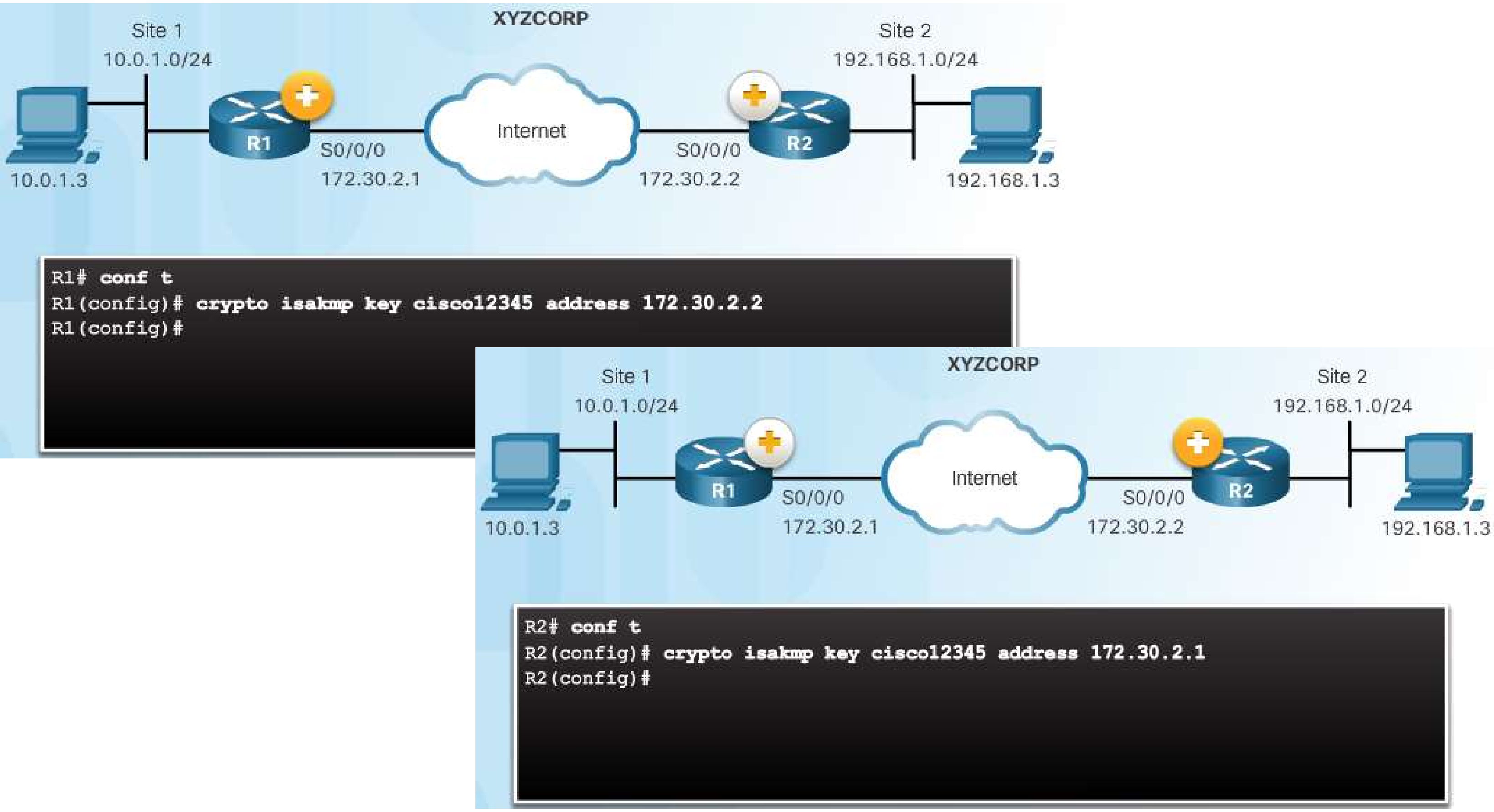
```
Router(config)#
crypto isakmp key keystring hostname peer-hostname
```



Parameter	Description
keystring	This parameter specifies the PSK. Use any combination of alphanumeric characters up to 128 bytes. This PSK must be identical on both peers.
peer-address	This parameter specifies the IP address of the remote peer.
hostname	This parameter specifies the hostname of the remote peer. This is the peer hostname concatenated with its domain name (for example, myhost.domain.com).

Configuring a Pre-Shared Key

Pre-Shared Key Configuration

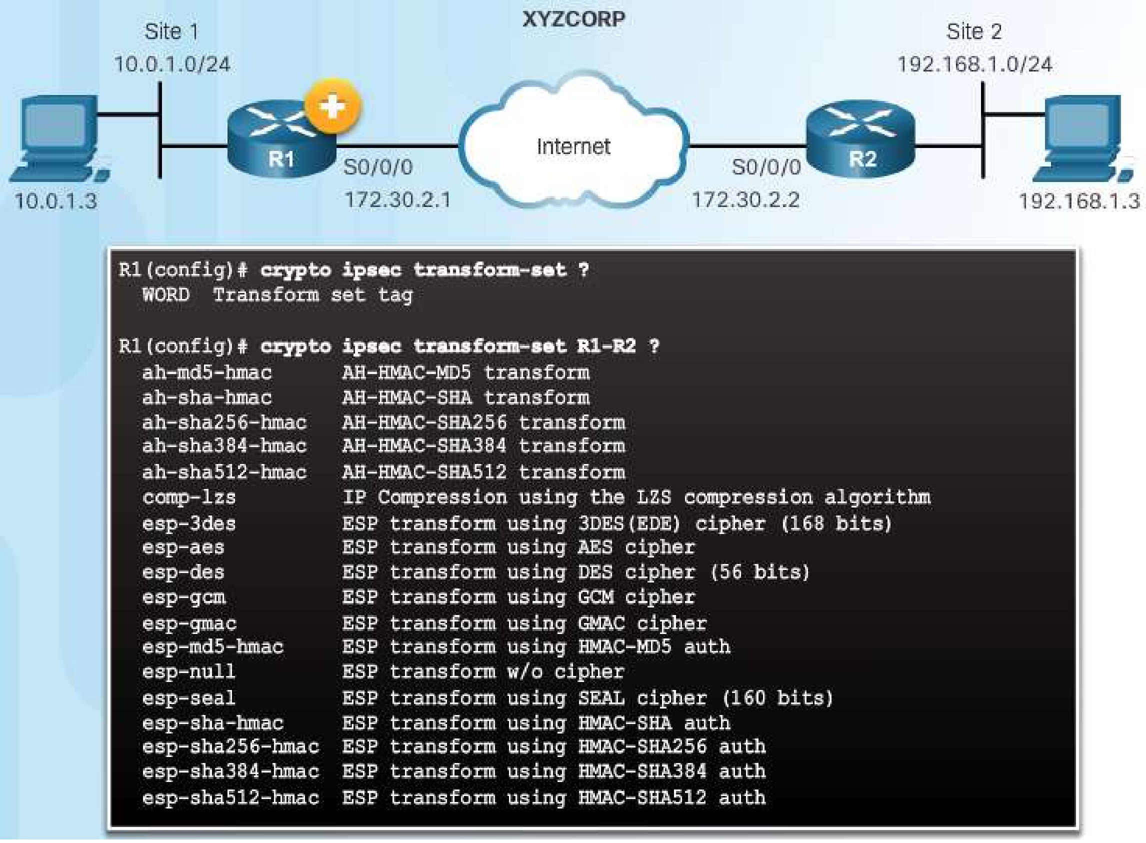


```
R1# conf t
R1(config)# crypto isakmp key cisco12345 address 172.30.2.2
R1(config)#
```

```
R2# conf t
R2(config)# crypto isakmp key cisco12345 address 172.30.2.1
R2(config)#
```

Configure IPsec Transform Set

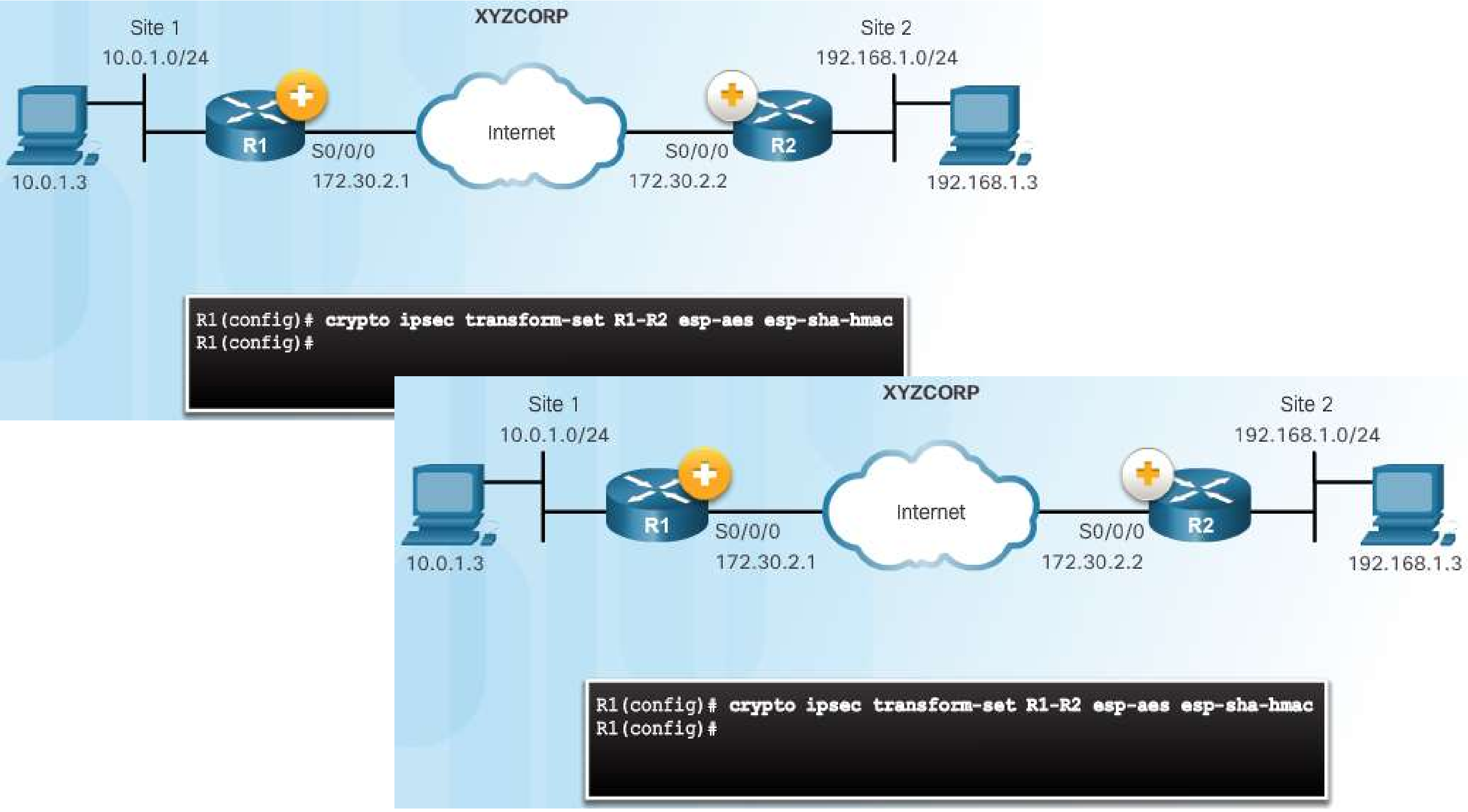
The `crypto ipsec transform-set` Command



```
R1(config)# crypto ipsec transform-set ?
  WORD  Transform set tag

R1(config)# crypto ipsec transform-set R1-R2 ?
  ah-md5-hmac      AH-HMAC-MD5 transform
  ah-sha-hmac      AH-HMAC-SHA transform
  ah-sha256-hmac   AH-HMAC-SHA256 transform
  ah-sha384-hmac   AH-HMAC-SHA384 transform
  ah-sha512-hmac   AH-HMAC-SHA512 transform
  comp-lzs         IP Compression using the LZS compression algorithm
  esp-3des         ESP transform using 3DES(EDE) cipher (168 bits)
  esp-aes          ESP transform using AES cipher
  esp-des          ESP transform using DES cipher (56 bits)
  esp-gcm          ESP transform using GCM cipher
  esp-gmac         ESP transform using GMAC cipher
  esp-md5-hmac     ESP transform using HMAC-MD5 auth
  esp-null         ESP transform w/o cipher
  esp-seal         ESP transform using SEAL cipher (160 bits)
  esp-sha-hmac     ESP transform using HMAC-SHA auth
  esp-sha256-hmac  ESP transform using HMAC-SHA256 auth
  esp-sha384-hmac  ESP transform using HMAC-SHA384 auth
  esp-sha512-hmac  ESP transform using HMAC-SHA512 auth
```

Configure IPsec Transform Set (Cont.)

The `crypto ipsec transform-set` Command

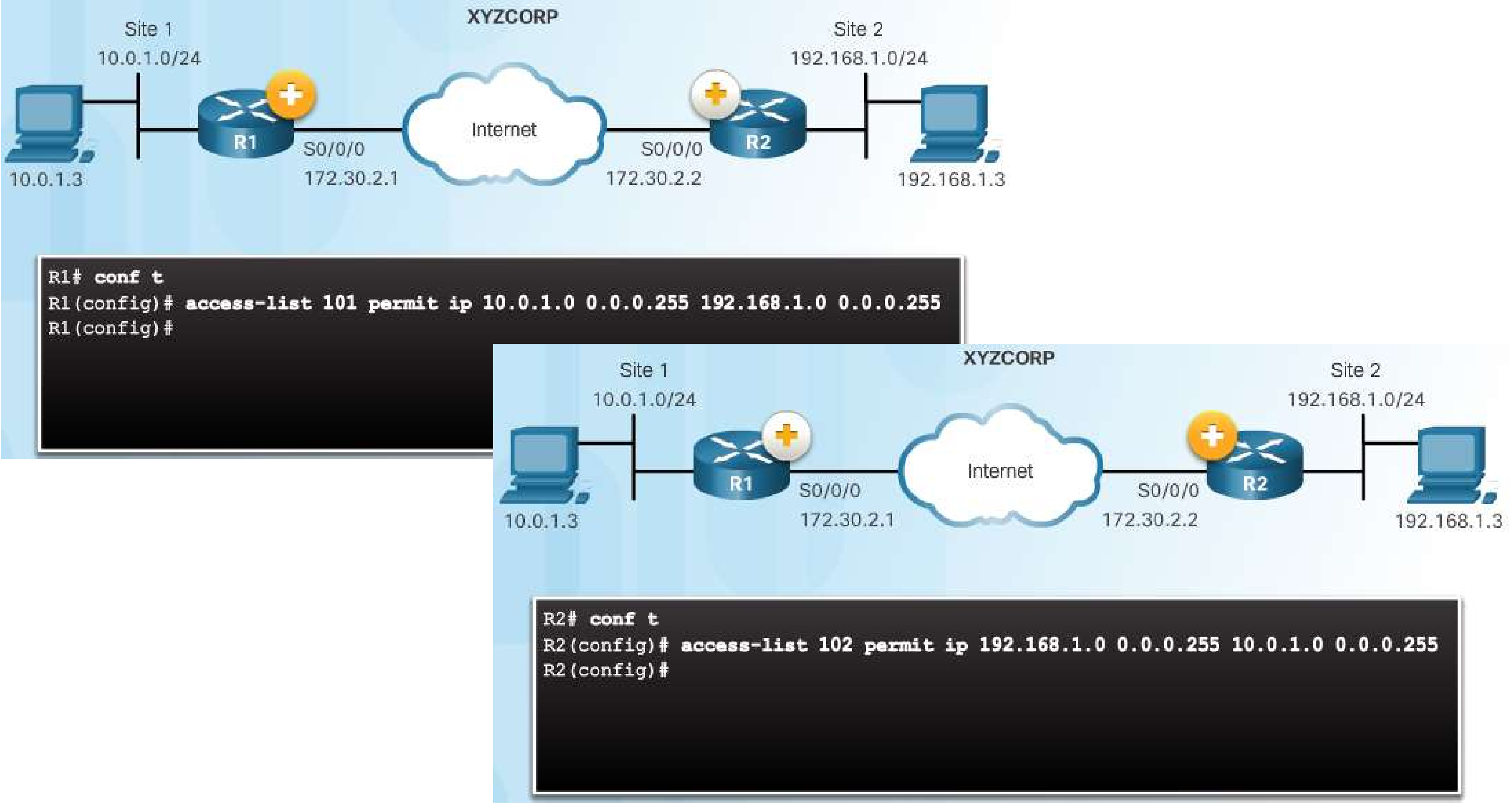


```
R1(config)# crypto ipsec transform-set R1-R2 esp-aes esp-sha-hmac
R1(config)#
```

```
R1(config)# crypto ipsec transform-set R1-R2 esp-aes esp-sha-hmac
R1(config)#
```

Configure the Crypto ACLs

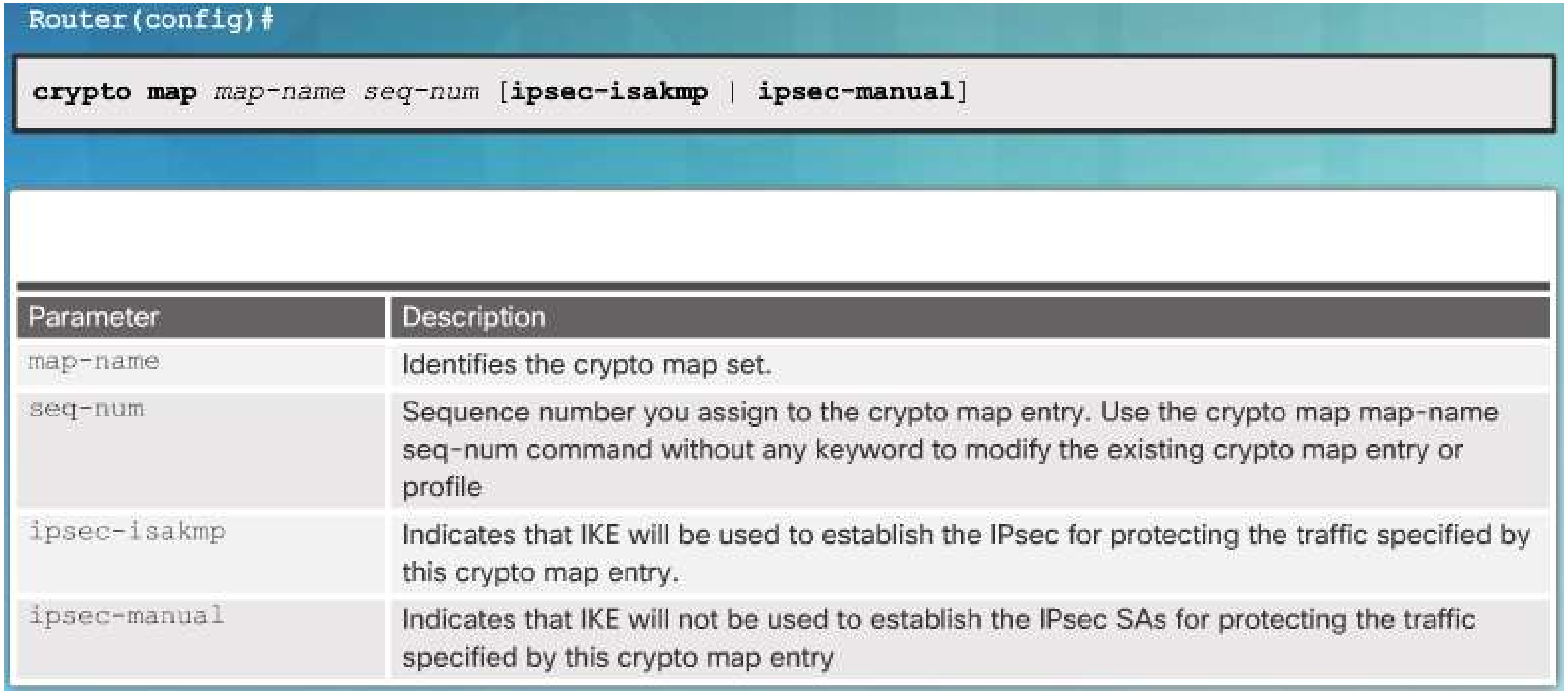
Configure an ACL to Define Interesting Traffic



```
R1# conf t
R1(config)# access-list 101 permit ip 10.0.1.0 0.0.0.255 192.168.1.0 0.0.0.255
R1(config)#
```

```
R2# conf t
R2(config)# access-list 102 permit ip 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
R2(config)#
```

Syntax to Configure a Crypto Map

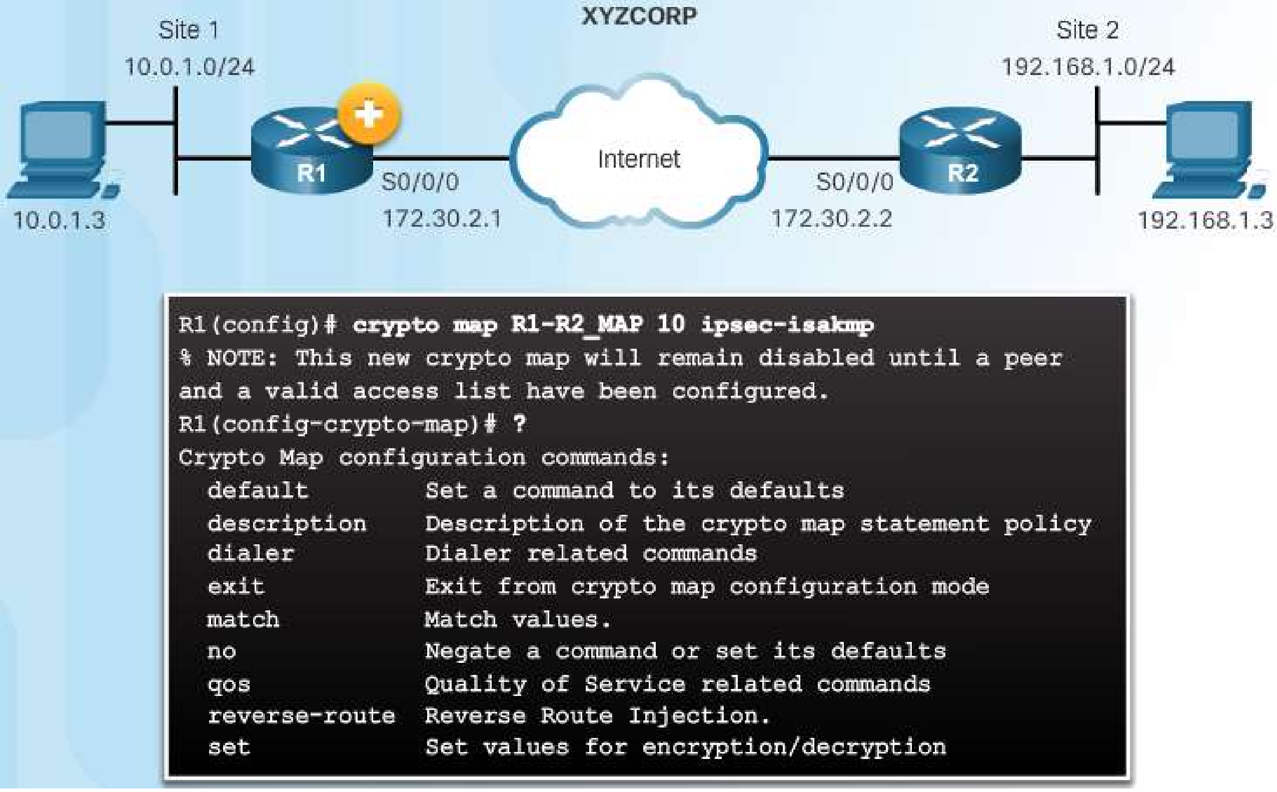


```
Router(config)#
crypto map map-name seq-num [ipsec-isakmp | ipsec-manual]
```

Parameter	Description
`map-name`	Identifies the crypto map set.
`seq-num`	Sequence number you assign to the crypto map entry. Use the crypto map map-name seq-num command without any keyword to modify the existing crypto map entry or profile
`ipsec-isakmp`	Indicates that IKE will be used to establish the IPsec for protecting the traffic specified by this crypto map entry.
`ipsec-manual`	Indicates that IKE will not be used to establish the IPsec SAs for protecting the traffic specified by this crypto map entry

Configure a Crypto Map

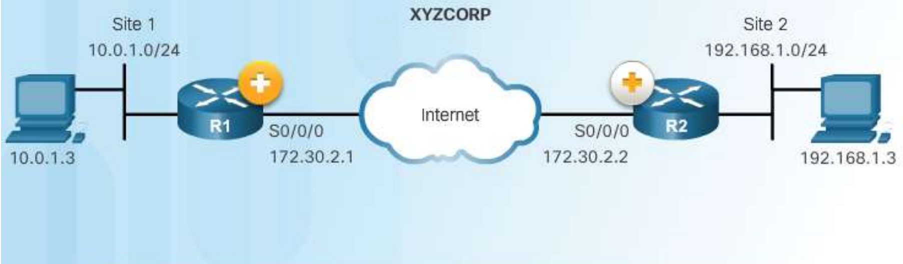
Crypto Map Configuration Commands



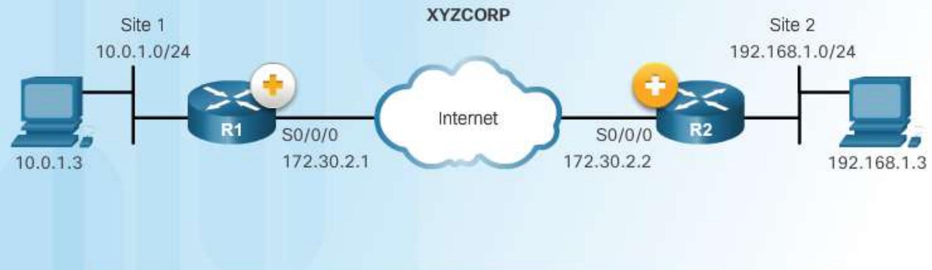
```
R1(config)# crypto map R1-R2_MAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
and a valid access list have been configured.
R1(config-crypto-map)# ?
Crypto Map configuration commands:
  default        Set a command to its defaults
  description    Description of the crypto map statement policy
  dialer         Dialer related commands
  exit           Exit from crypto map configuration mode
  match          Match values.
  no             Negate a command or set its defaults
  qos            Quality of Service related commands
  reverse-route  Reverse Route Injection.
  set            Set values for encryption/decryption
```

XYZCORP Crypto Map Configuration

Crypto Map Configuration:



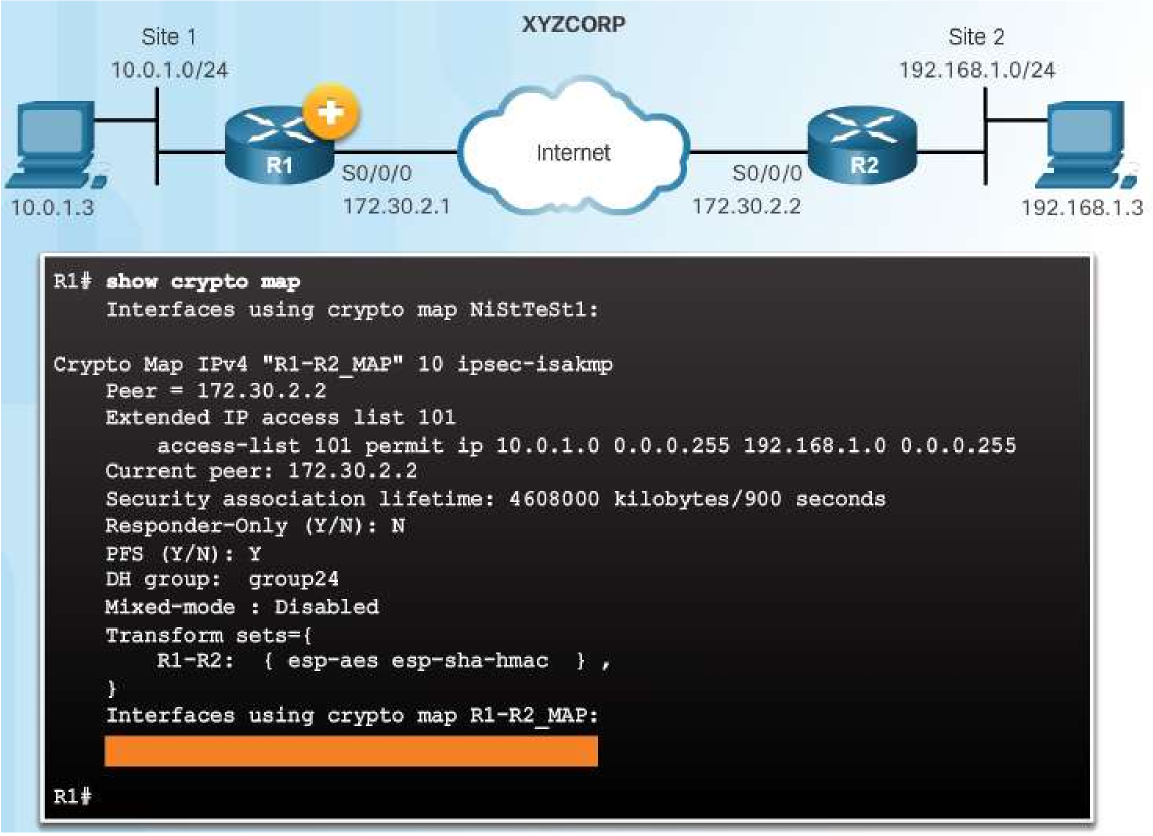
```
R1(config)# crypto map R1-R2_MAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
and a valid access list have been configured.
R1(config-crypto-map)# match address 101
R1(config-crypto-map)# set transform-set R1-R2
R1(config-crypto-map)# set peer 172.30.2.2
R1(config-crypto-map)# set pfs group24
R1(config-crypto-map)# set security-association lifetime seconds 900
R1(config-crypto-map)# exit
R1(config)#
```



```
R2(config)# crypto map R1-R2_MAP 10 ipsec-isakmp
% NOTE: This new crypto map will remain disabled until a peer
and a valid access list have been configured.
R2(config-crypto-map)# match address 102
R2(config-crypto-map)# set transform-set R1-R2
R2(config-crypto-map)# set peer 172.30.2.1
R2(config-crypto-map)# set pfs group24
R2(config-crypto-map)# set security-association lifetime seconds 900
R2(config-crypto-map)# exit
R2(config)#
```

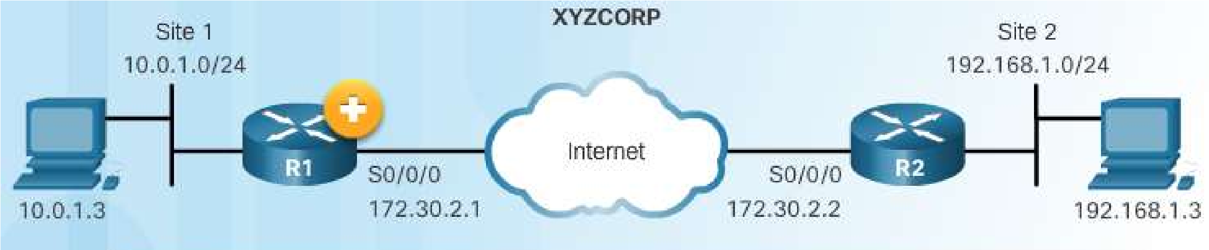
XYZCORP Crypto Map Configuration

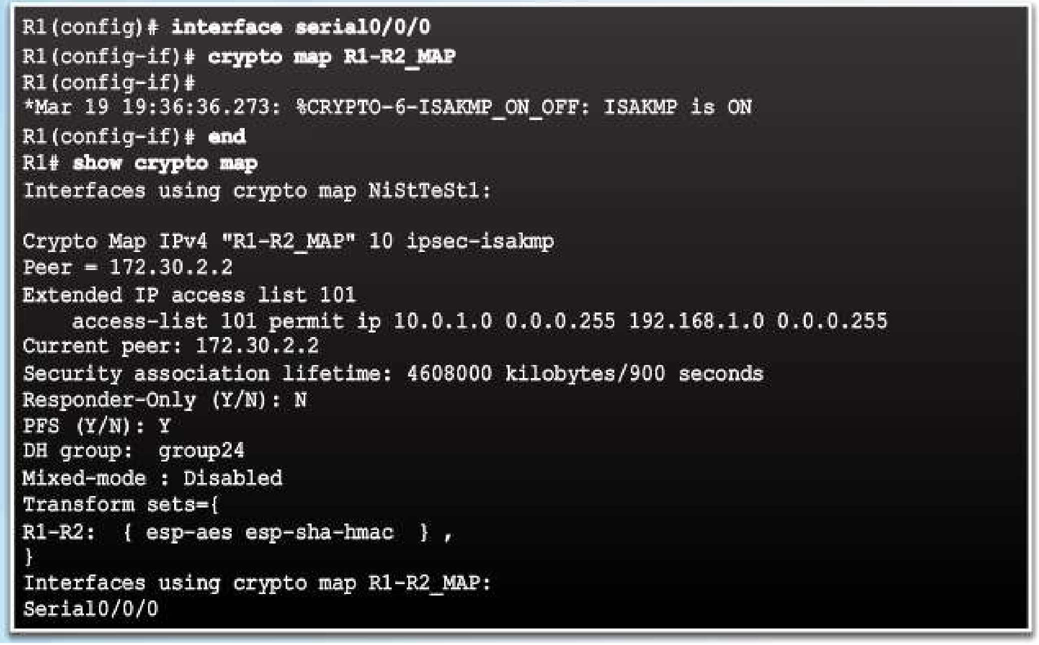
Crypto Map Configuration:



```
R1# show crypto map
    Interfaces using crypto map NiStTeSt1:

Crypto Map IPv4 "R1-R2_MAP" 10 ipsec-isakmp
    Peer = 172.30.2.2
    Extended IP access list 101
        access-list 101 permit ip 10.0.1.0 0.0.0.255 192.168.1.0 0.0.0.255
    Current peer: 172.30.2.2
    Security association lifetime: 4608000 kilobytes/900 seconds
    Responder-Only (Y/N): N
    PFS (Y/N): Y
    DH group:  group24
    Mixed-mode : Disabled
    Transform sets={
        R1-R2:  { esp-aes esp-sha-hmac  } ,
    }
    Interfaces using crypto map R1-R2_MAP:


R1#
```

Apply the Crypto Map

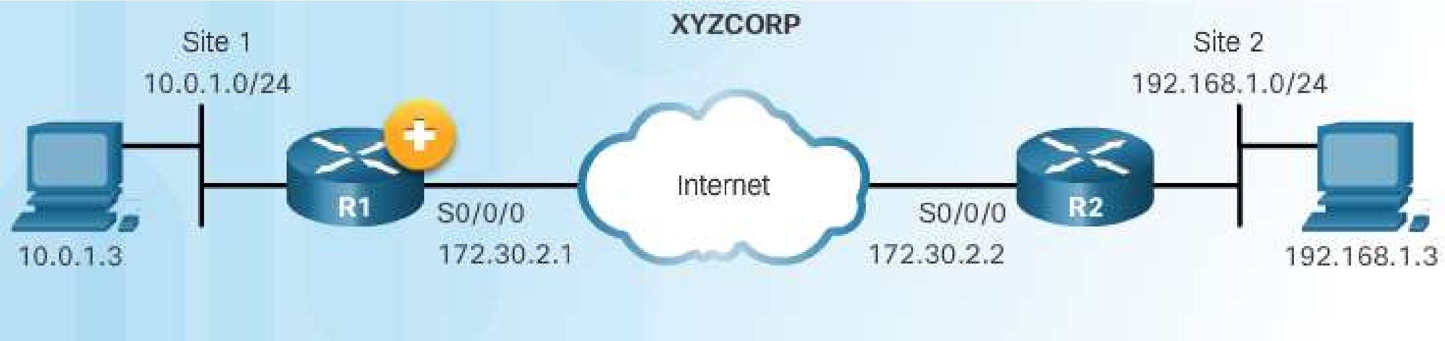


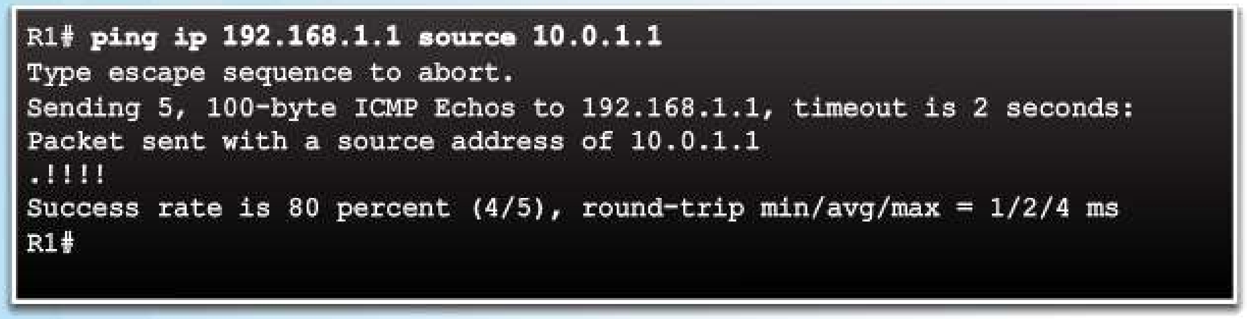


```
R1(config)# interface serial0/0/0
R1(config-if)# crypto map R1-R2_MAP
R1(config-if)#
*Mar 19 19:36:36.273: %CRYPTO-6-ISAKMP_ON_OFF: ISAKMP is ON
R1(config-if)# end
R1# show crypto map
Interfaces using crypto map NiStTeSt1:

Crypto Map IPv4 "R1-R2_MAP" 10 ipsec-isakmp
Peer = 172.30.2.2
Extended IP access list 101
    access-list 101 permit ip 10.0.1.0 0.0.0.255 192.168.1.0 0.0.0.255
Current peer: 172.30.2.2
Security association lifetime: 4608000 kilobytes/900 seconds
Responder-Only (Y/N): N
PFS (Y/N): Y
DH group:  group24
Mixed-mode : Disabled
Transform sets={
R1-R2:  { esp-aes esp-sha-hmac  } ,
}
Interfaces using crypto map R1-R2_MAP:
Serial0/0/0
```

Send Interesting Traffic

Use Extended Ping to Send Interesting Traffic

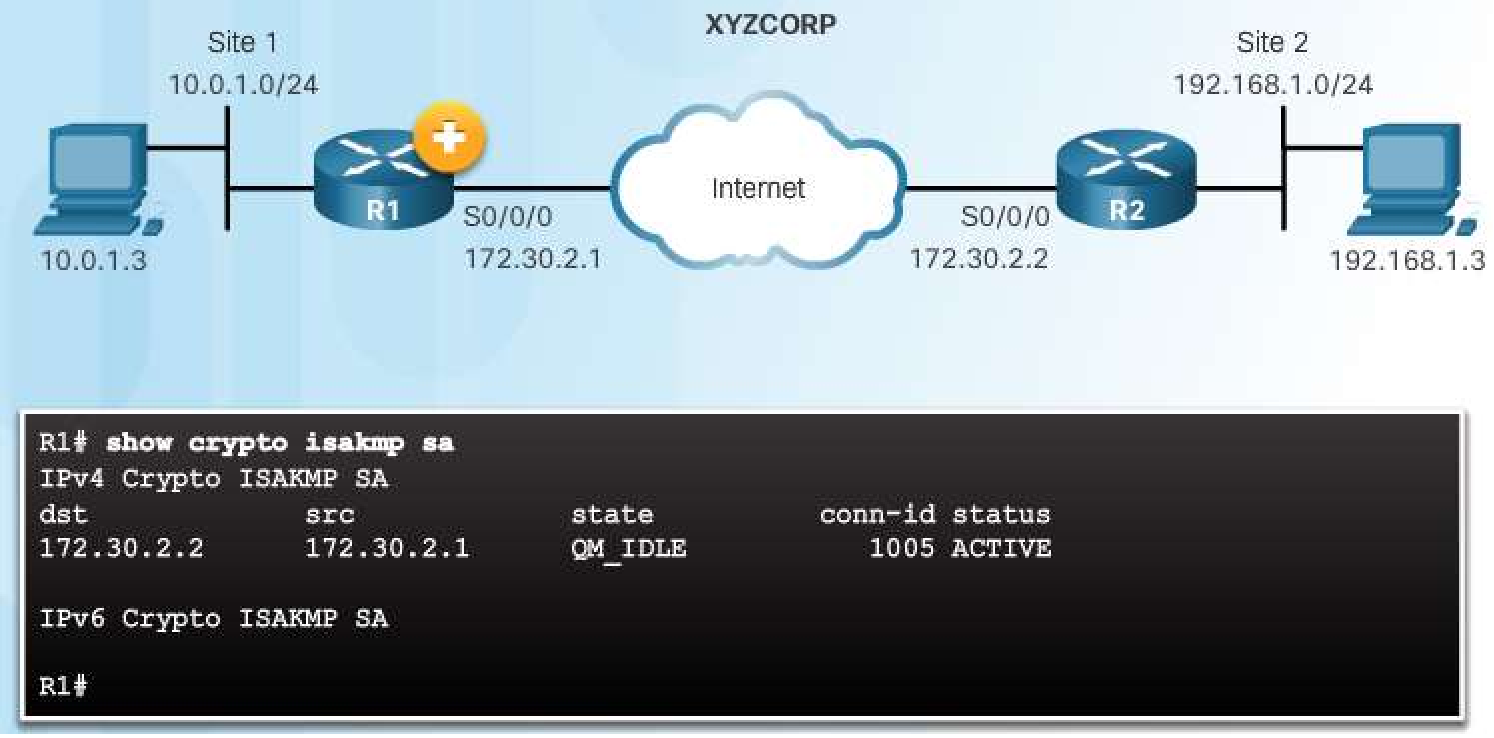




```
R1# ping ip 192.168.1.1 source 10.0.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.0.1.1
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/2/4 ms
R1#
```

Verify ISAKMP and IPsec Tunnels

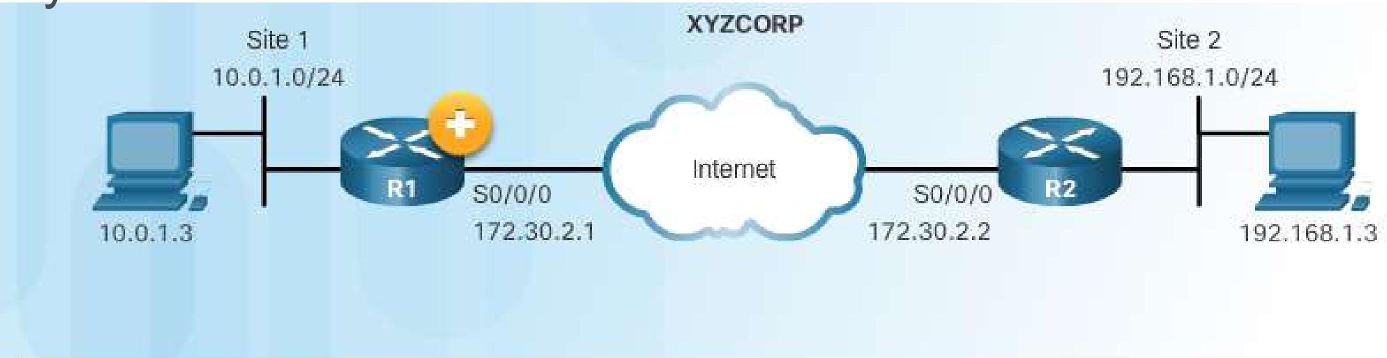
Verify the ISAKMP Tunnel is Established

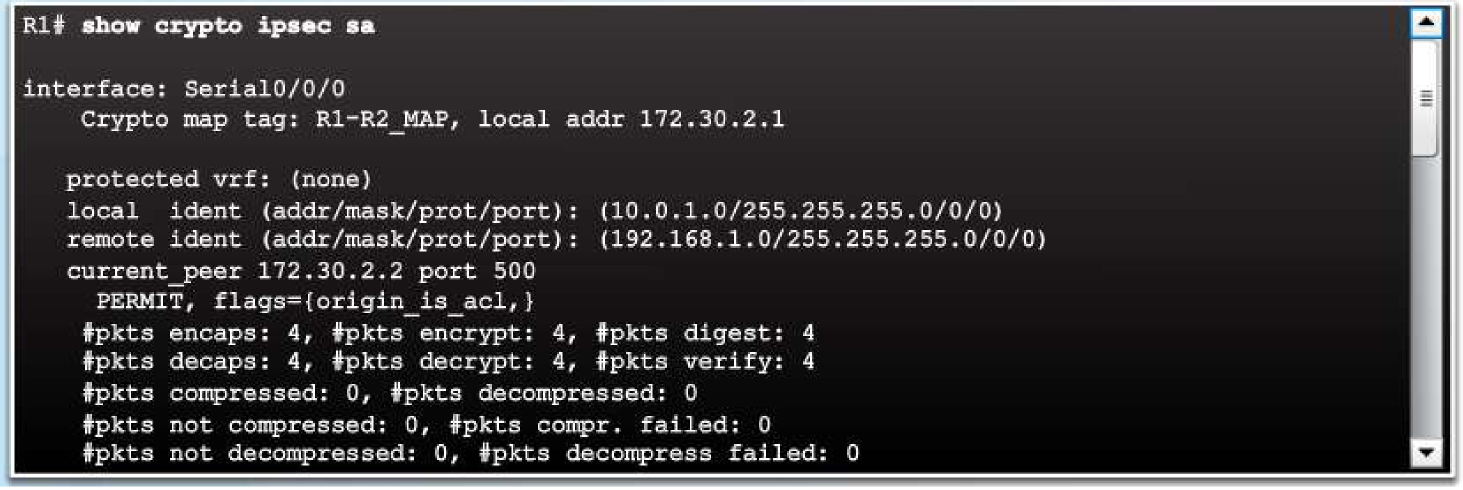


```
R1# show crypto isakmp sa
IPv4 Crypto ISAKMP SA
dst             src             state          conn-id status
172.30.2.2      172.30.2.1      QM_IDLE           1005 ACTIVE

IPv6 Crypto ISAKMP SA

R1#
```

Verify ISAKMP and IPsec Tunnels

Verify the IPsec Tunnel is Established





```
R1# show crypto ipsec sa

interface: Serial0/0/0
    Crypto map tag: R1-R2_MAP, local addr 172.30.2.1

   protected vrf: (none)
   local  ident (addr/mask/prot/port): (10.0.1.0/255.255.255.0/0/0)
   remote ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
   current_peer 172.30.2.2 port 500
     PERMIT, flags={origin_is_acl,}
    #pkts encaps: 4, #pkts encrypt: 4, #pkts digest: 4
    #pkts decaps: 4, #pkts decrypt: 4, #pkts verify: 4
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
```