ISP and NSP Security Workshop

APRICOT 2013
Day 4
Core Security

Cryptosystem

- A system to accomplish the encryption/decryption, user authentication, hashing, and key-exchange processes.
- A cryptosystem may use one of several different methods, depending on the policy intended for various user traffic situations.

Encryption / Decryption

- Encryption transforms information (clear text) into ciphertext which is not readable by unauthorized users.
- Decryption transforms ciphertext back into clear text making it readable by authorized users.
- Popular encryption algorithms include:
  - DES
  - 3DES
  - AES

Authentication / Hashing

- Guarantees message integrity by using an algorithm to convert a variable length message and shared secret key into a single fixed-length string.
- Popular hashing methods include:
  - SHA (Cisco default)
  - MD5

Non-repudiation

- Is the ability to prove a transaction occurred.
  - Similar to a signed package received from a shipping company.
- This is very important in financial transactions and similar data transactions.

Diffie-Hellman Key Exchange

- How do the encrypting and decrypting devices get the shared secret key?
  - The easiest method is Diffie-Hellman public key exchange.
- Used to create a shared secret key without prior knowledge.
- This secret key is required by:
  - The encryption algorithm (DES, 3DES, AES)
  - The authentication method (MD5 and SHA-1)

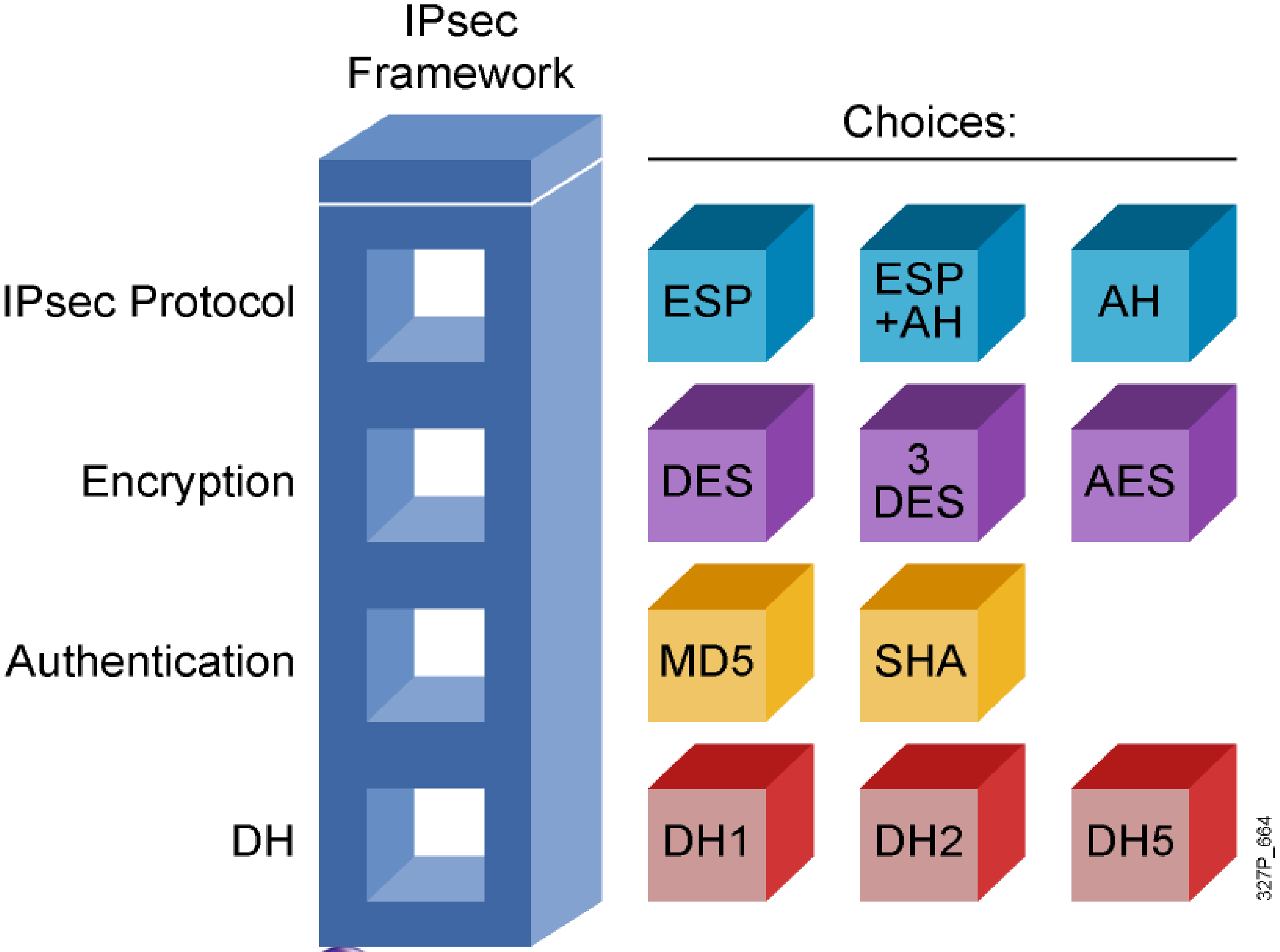
Pre-Shared Key

- Identifies a communicating party during a phase 1 IKE negotiation.
- The key must be pre-shared with another party before the peers routers can communicate.

IPsec - Internet Protocol Security

- A “framework” of open standards developed by the IETF to create a secure tunnel at the network (IP) layer.
  - It spells out the rules for secure communications.
- IPsec is not bound to any specific encryption or authentication algorithms, keying technology, or security algorithms.

IPsec Protocol Framework



IPsec Framework	Choices:		
IPsec Protocol	ESP	ESP +AH	AH
Encryption	DES	3 DES	AES
Authentication	MD5	SHA	
DH	DH1	DH2	DH5

Crypto Map

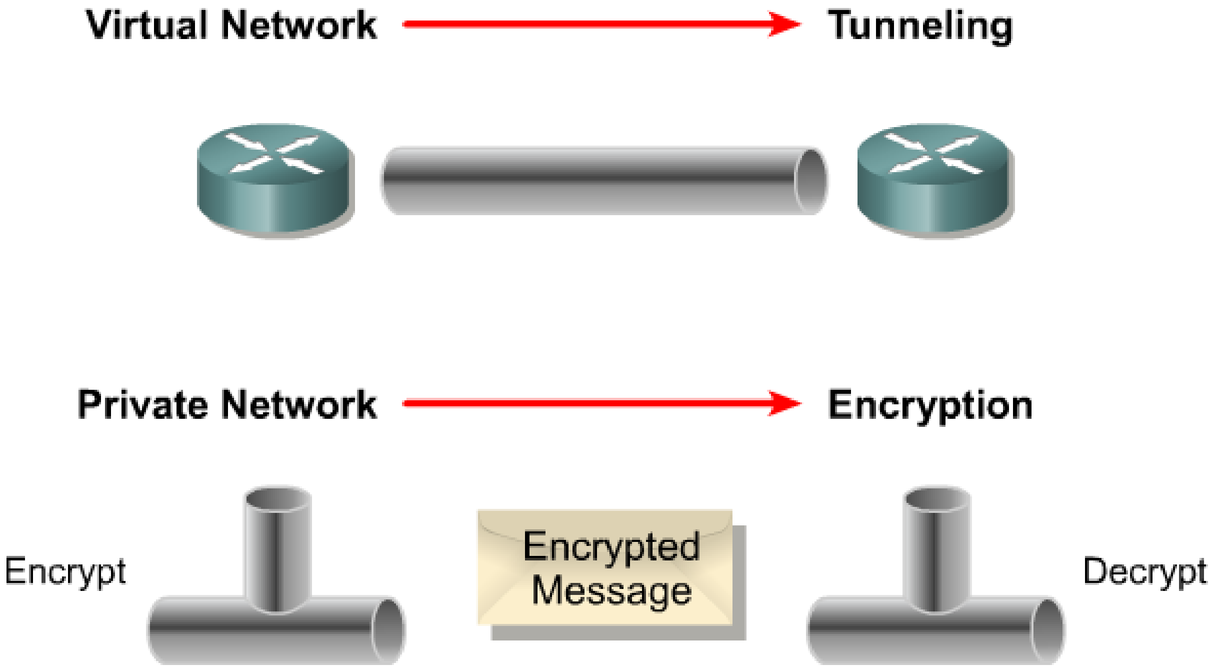
- A Cisco IOS software configuration entity that performs two primary functions.
  - First, it selects data flows that need security processing.
  - Second, it defines the policy for these flows and the crypto peer that traffic needs to go to.
- A crypto map is applied to an interface.

SA - Security Association

- Is a contract between two parties indicating what security parameters, such as keys and algorithms will be used.
- A Security Parameter Index (SPI) identifies each established SA.

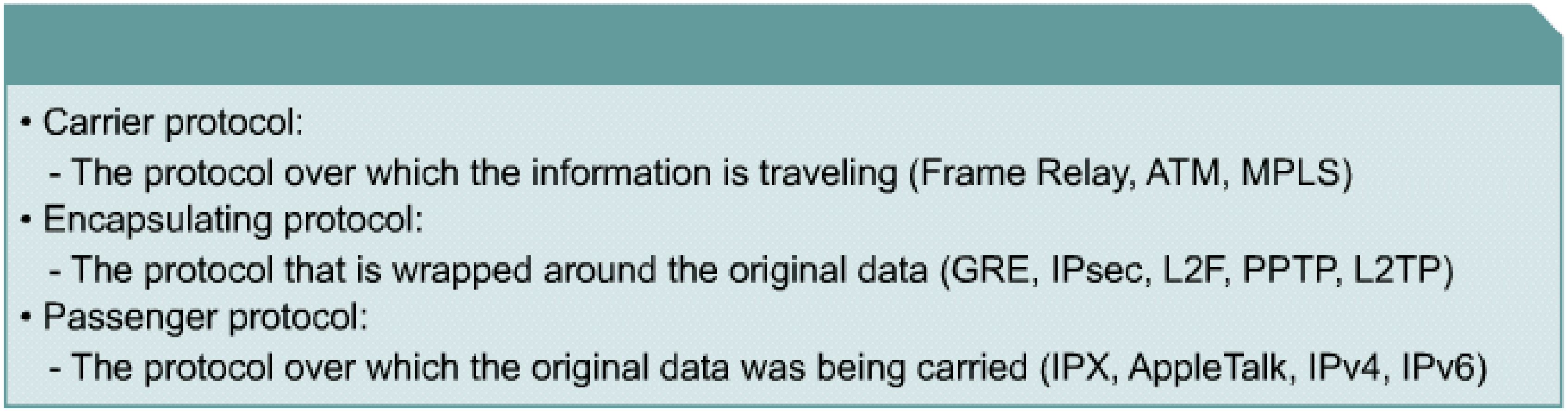
VPN Concepts

- A secure VPN is a combination of concepts:



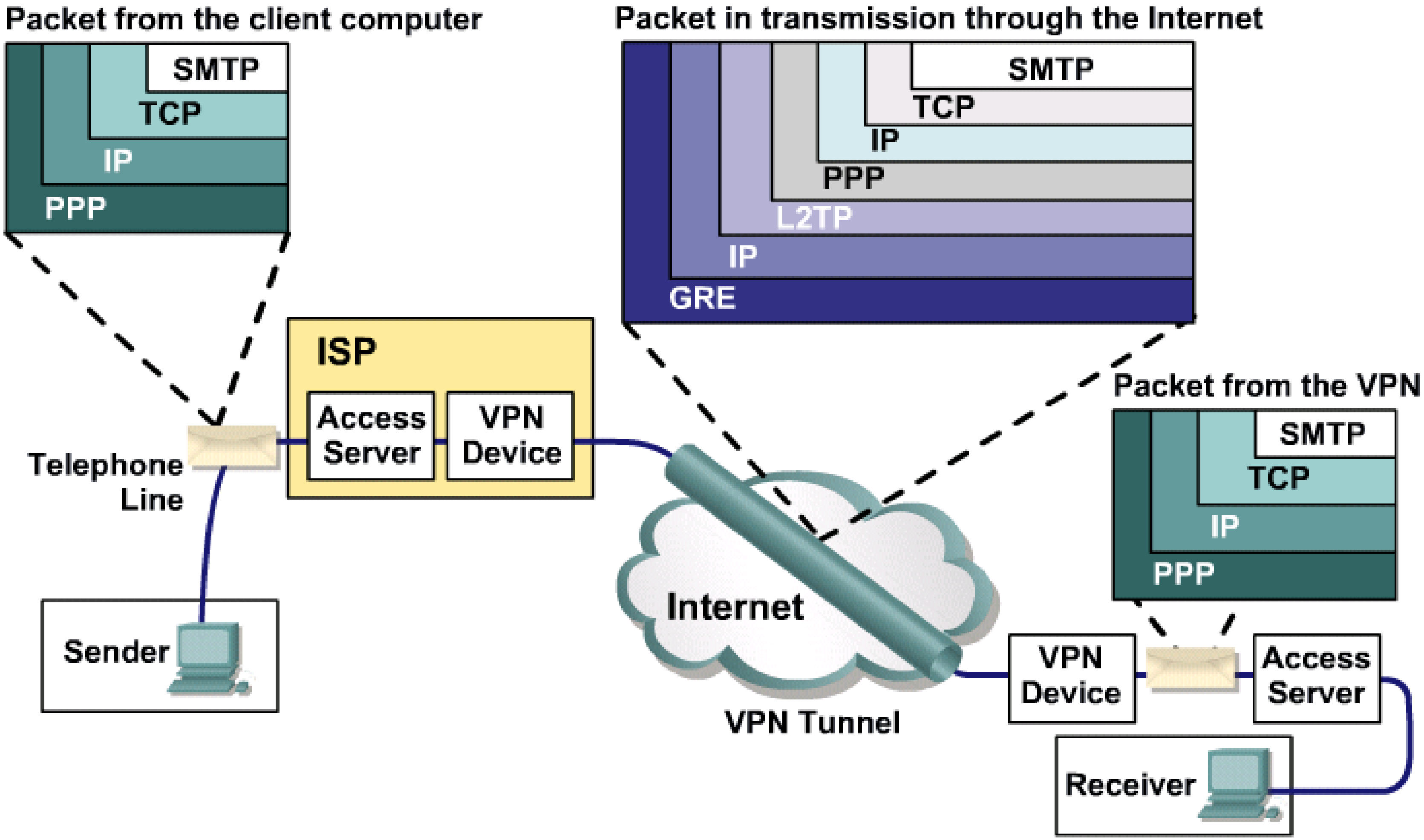
Virtual Private Network = Tunneling + Encryption

VPN Packet Encapsulation

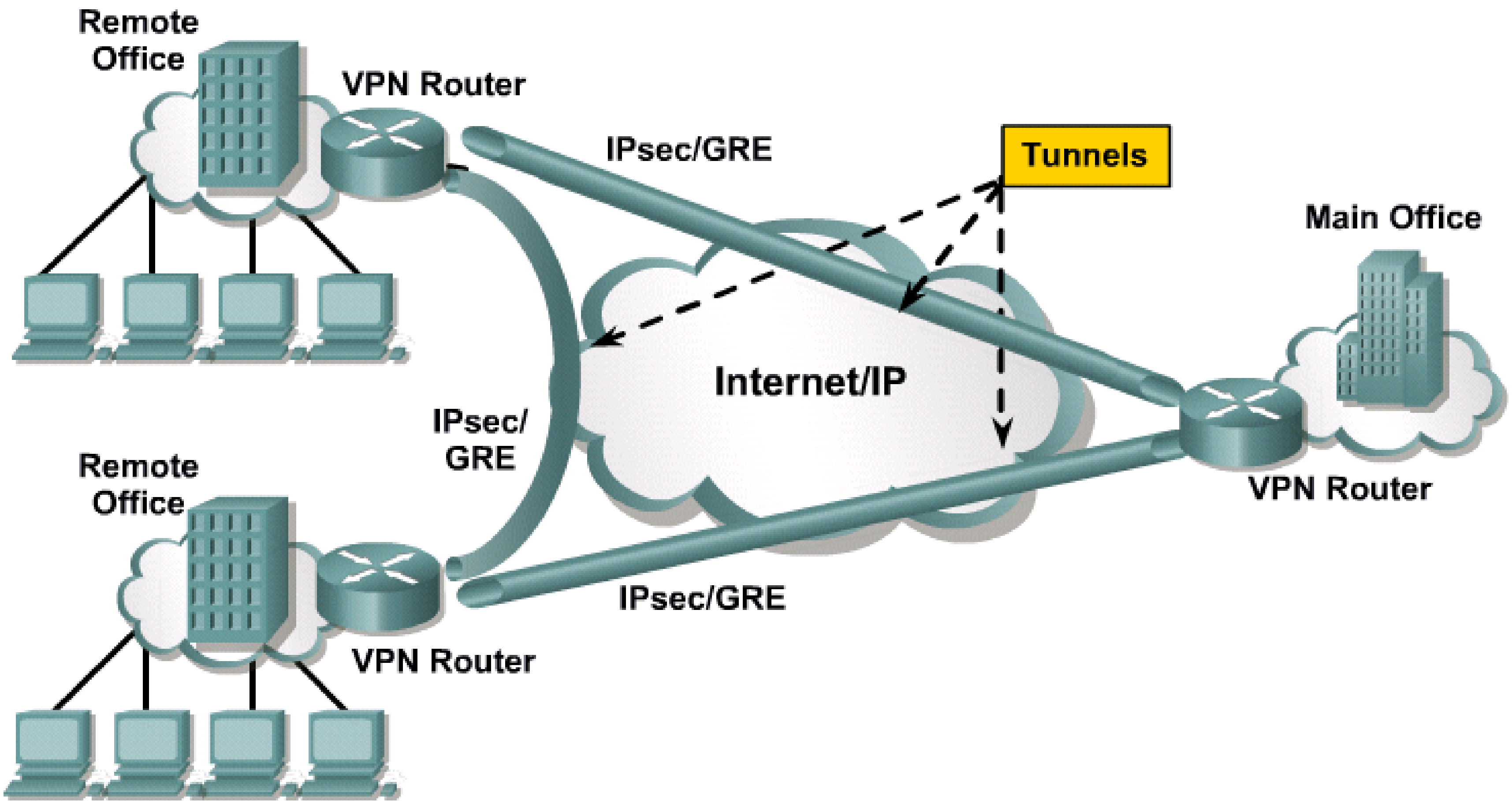


- Carrier protocol:
  - The protocol over which the information is traveling (Frame Relay, ATM, MPLS)
- Encapsulating protocol:
  - The protocol that is wrapped around the original data (GRE, IPsec, L2F, PPTP, L2TP)
- Passenger protocol:
  - The protocol over which the original data was being carried (IPX, AppleTalk, IPv4, IPv6)

VPN Packet Encapsulation



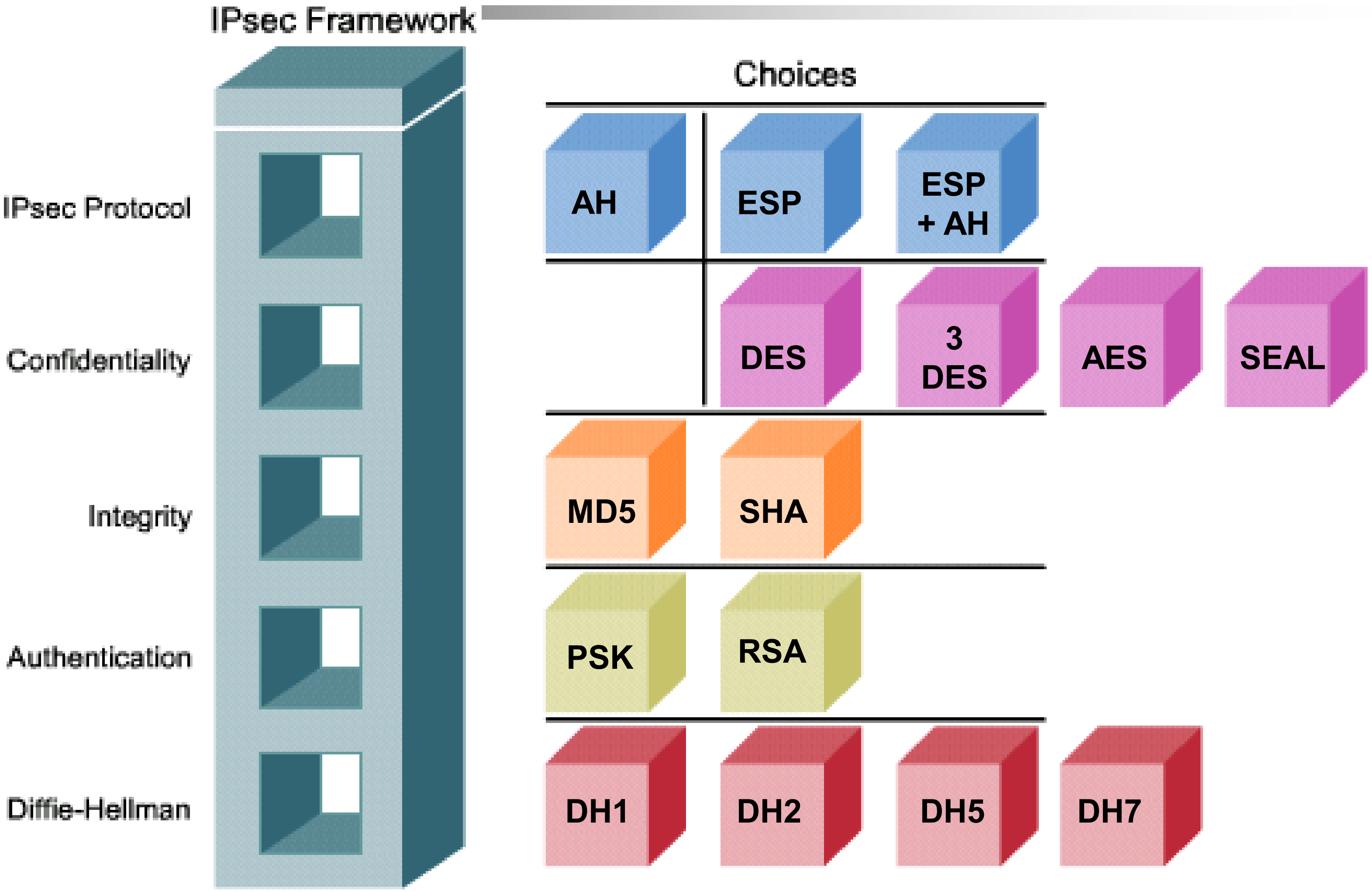
Site-to-Site VPNs



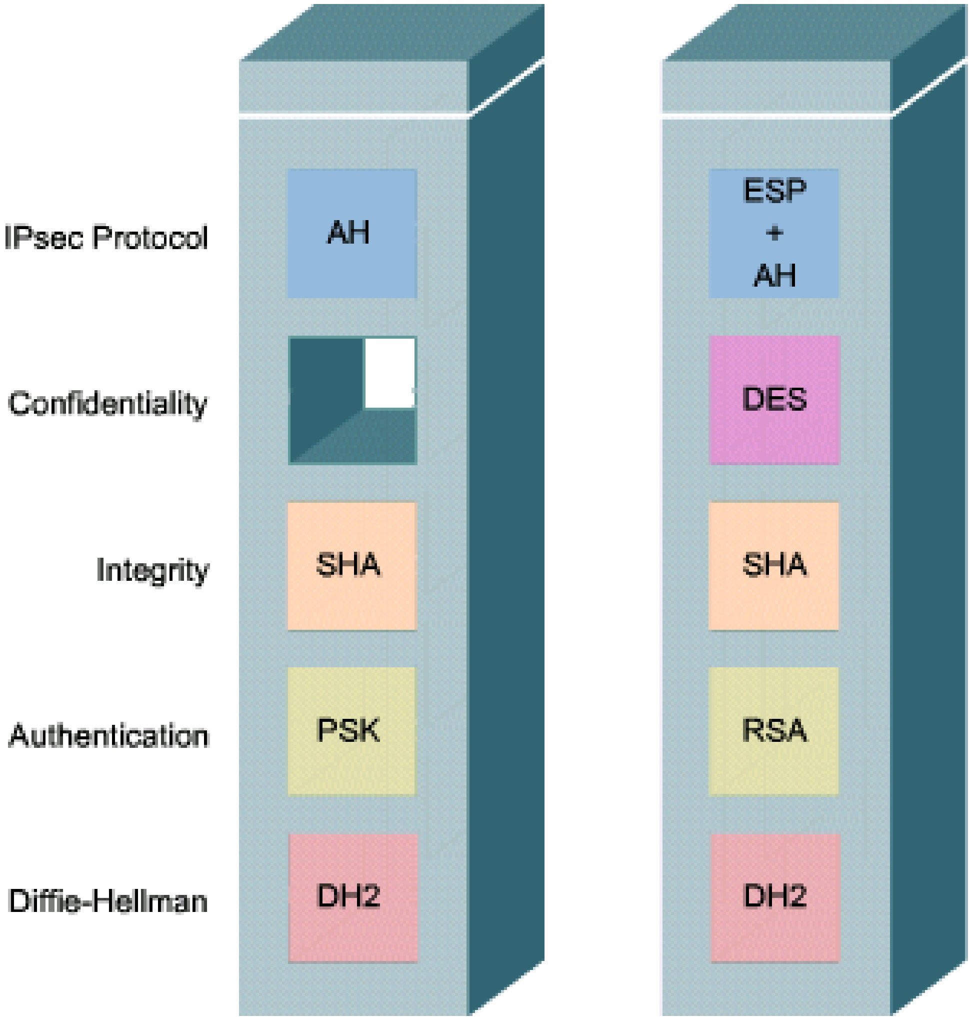
IPsec - Internet Protocol Security

- A “framework” of open standards developed by the IETF to create a secure tunnel at the network (IP) layer.
    - It spells out the rules for secure communications.
    - RFC 2401 - RFC 2412
- IPsec is not bound to any specific encryption or authentication algorithms, keying technology, or security algorithms.
- IPsec allows newer and better algorithms to be implemented without patching the existing IPsec standards.

IPsec Protocol Framework

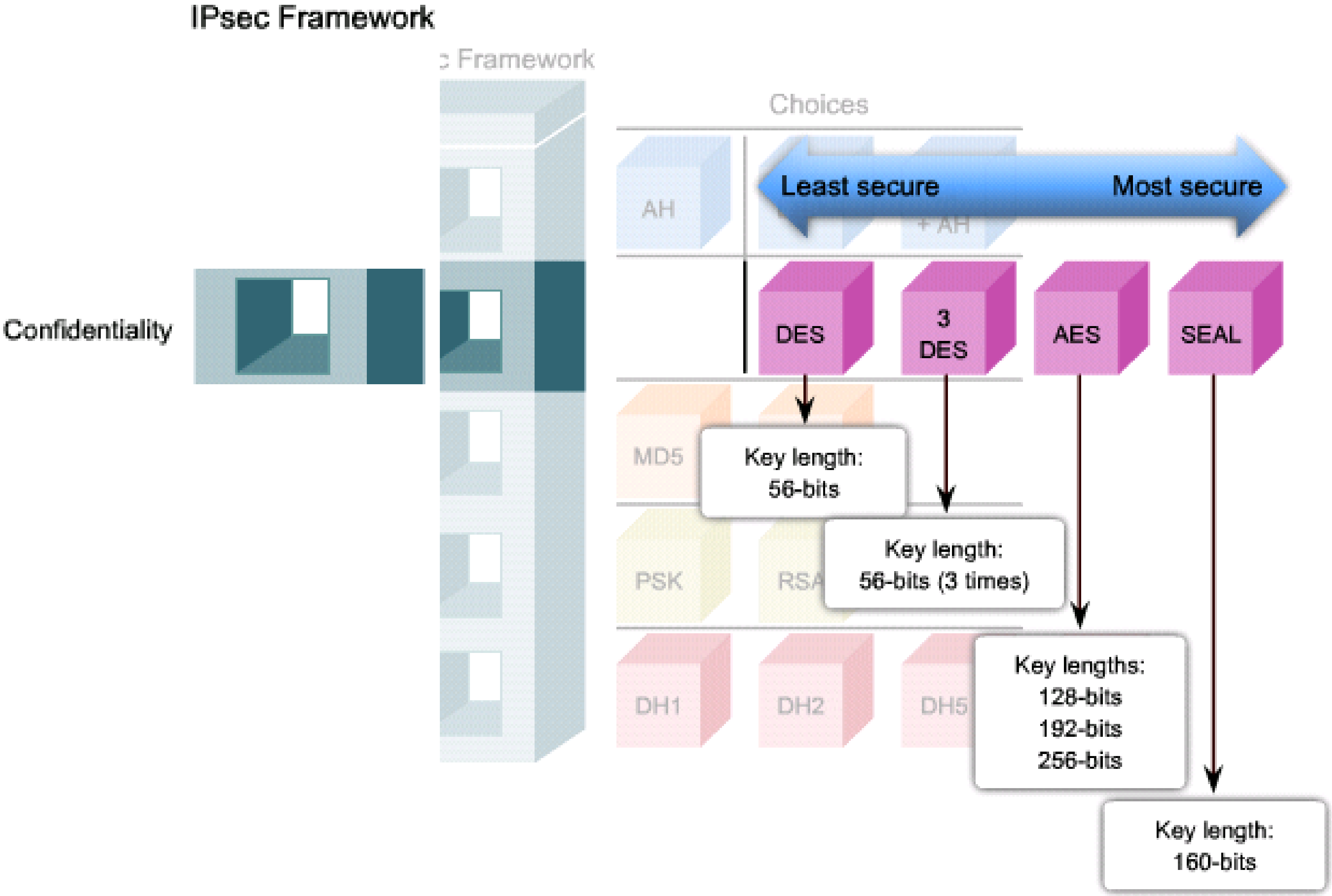


IPsec Protocol Framework



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

Confidentiality



IPsec Framework

Confidentiality

Least secure — Most secure

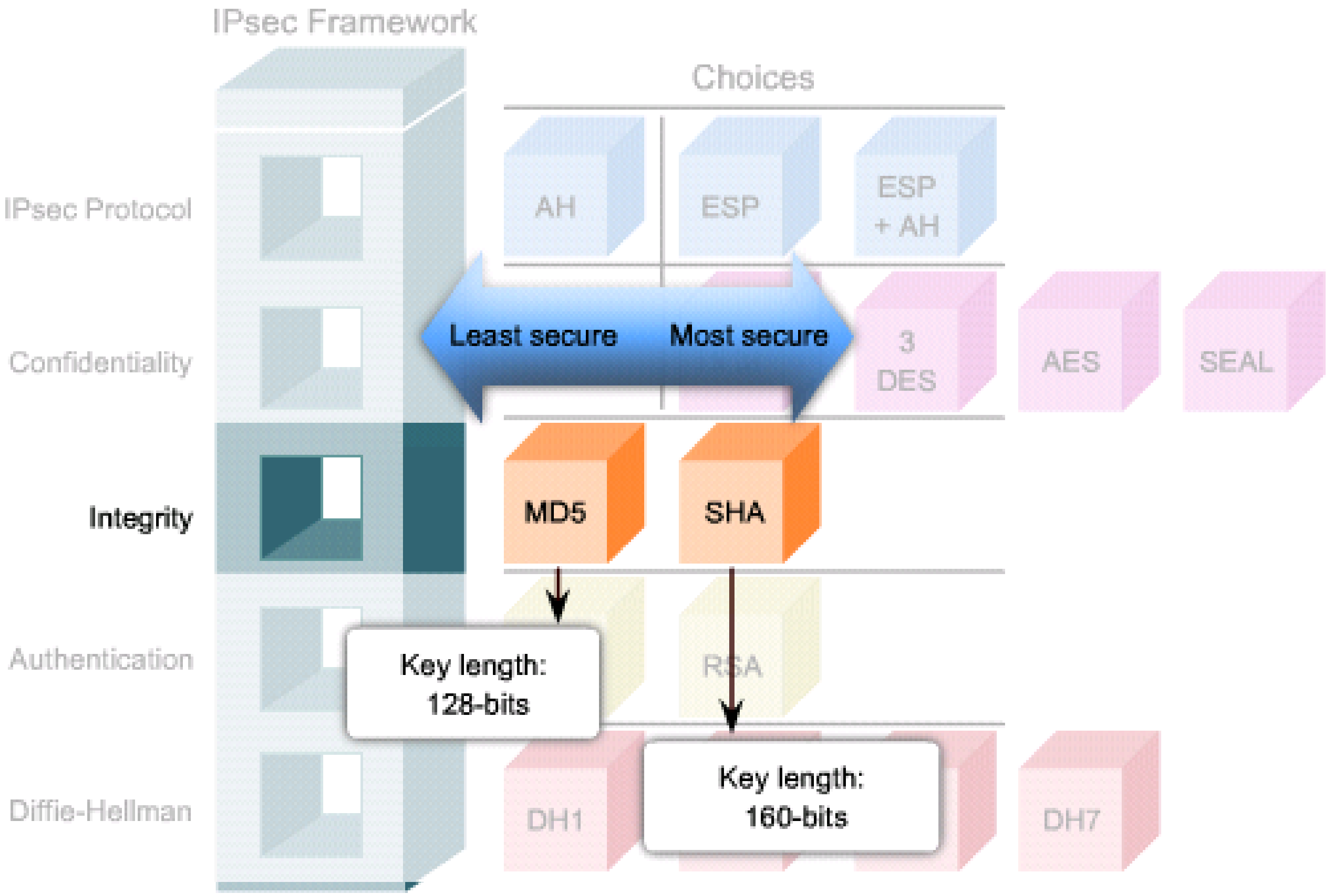
DES: Key length: 56-bits

3 DES: Key length: 56-bits (3 times)

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

SEAL: Key length: 160-bits

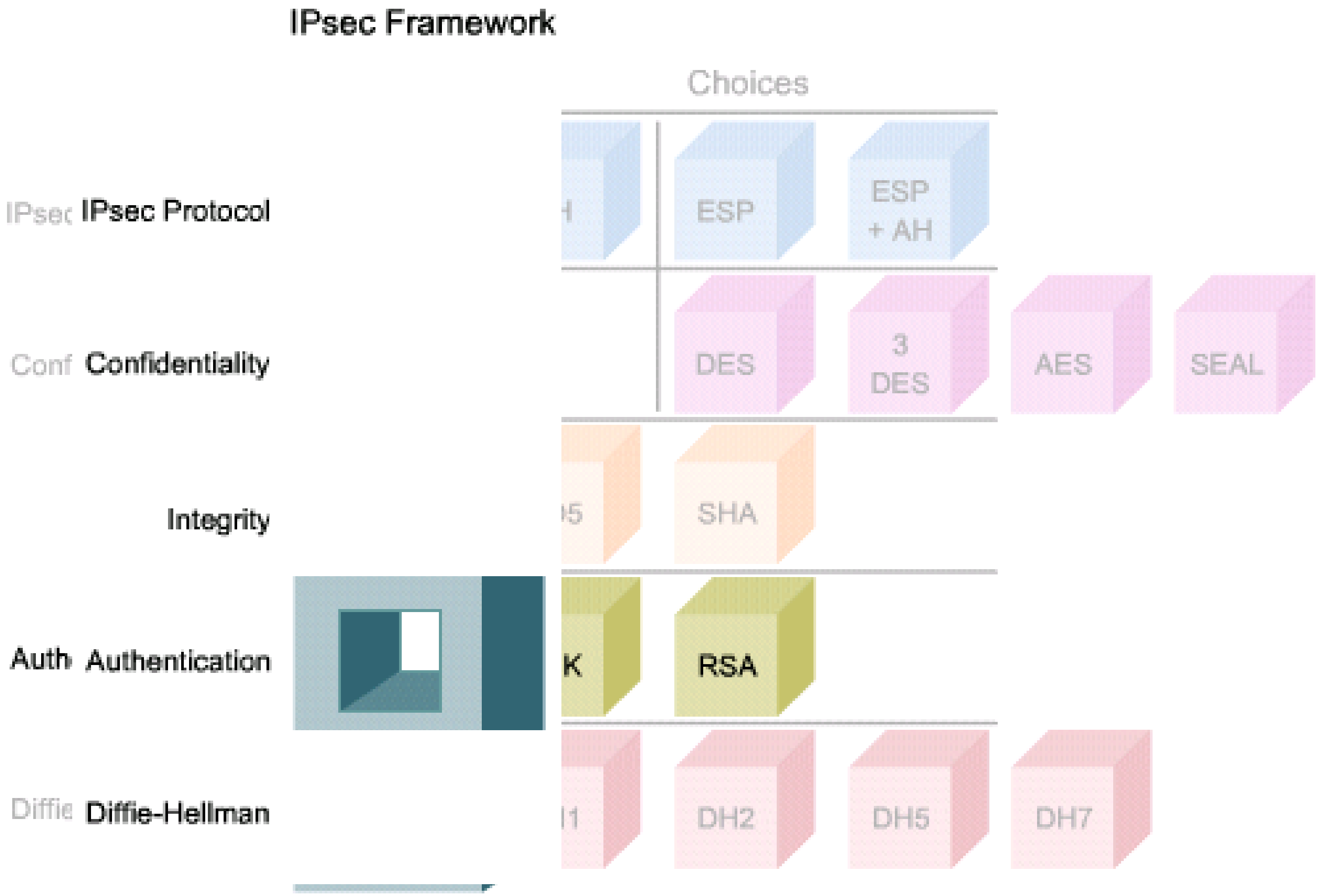
Integrity



Key length: 128-bits

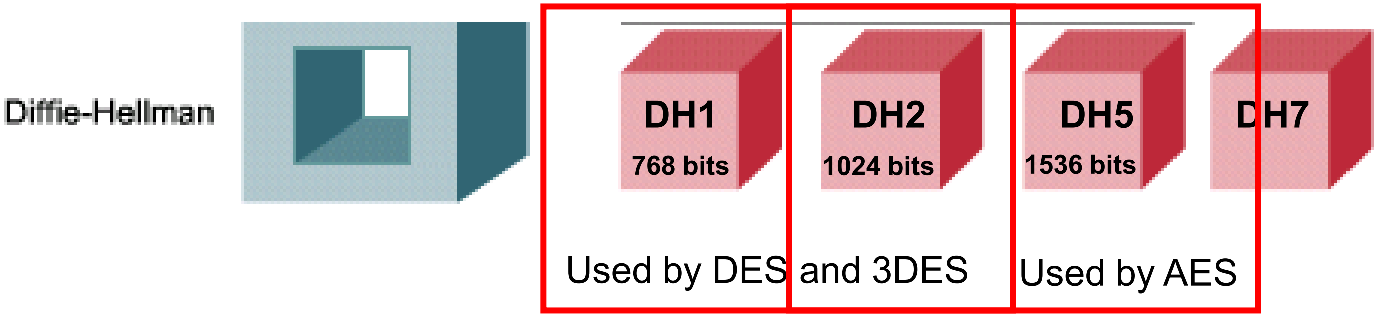
Key length: 160-bits

Authentication



Secure Key Exchange

IPsec Framework



Diffie-Hellman

DH1
768 bits

DH2
1024 bits

DH5
1536 bits

DH7

Used by DES and 3DES

Used by AES

IPsec Framework Protocols

- IPsec uses two main protocols to create a security framework:
  - AH: Authentication Header
  - ESP: Encapsulating Security Payload

Authentication Header (AH)

- AH provides authentication and optional replay-detection services.
  - It authenticates the sender of the data.
  - AH operates on protocol number 51.
  - AH supports the HMAC-MD5 and HMAC-SHA-1 algorithms.

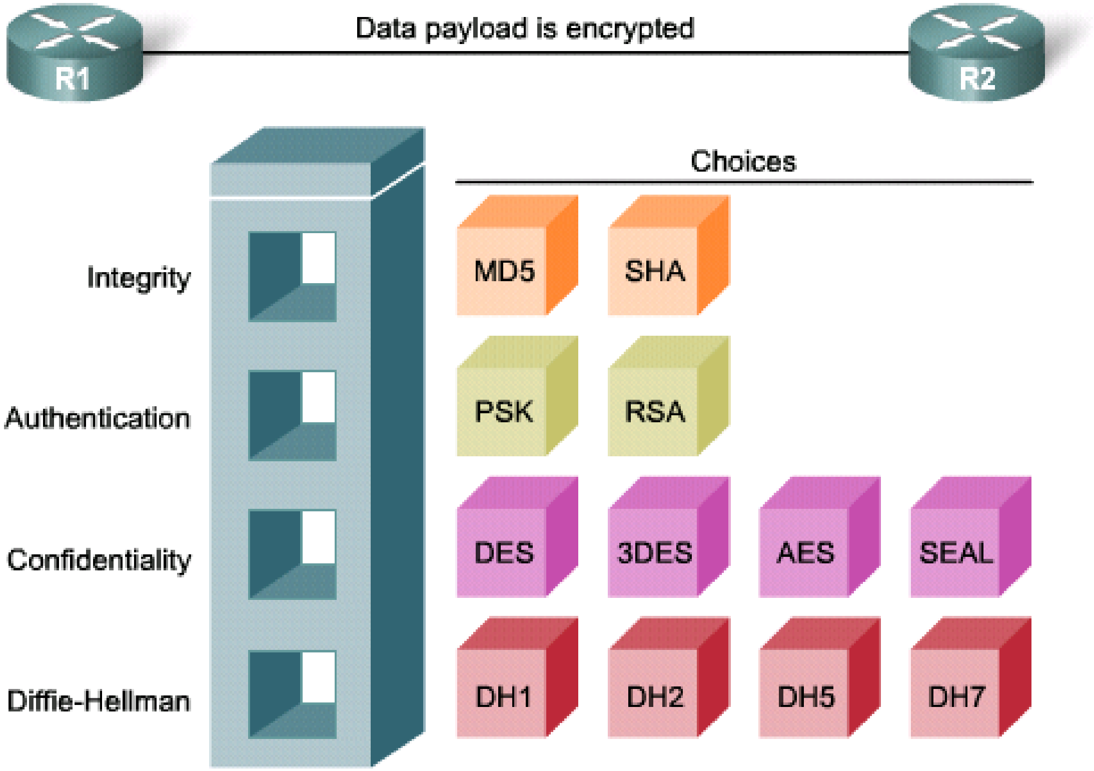
Authentication Header (AH)

- AH does not provide confidentiality (encryption).
  - It is appropriate to use when confidentiality is not required or permitted.
  - All text is transported unencrypted.
- It only ensures the origin of the data and verifies that the data has not been modified during transit.
- If the AH protocol is used alone, it provides weak protection.
- AH can have problems if the environment uses NAT.

Encapsulating Security Payload (ESP)

ESP provides the same security services as AH (authentication and integrity) AND encryption service.

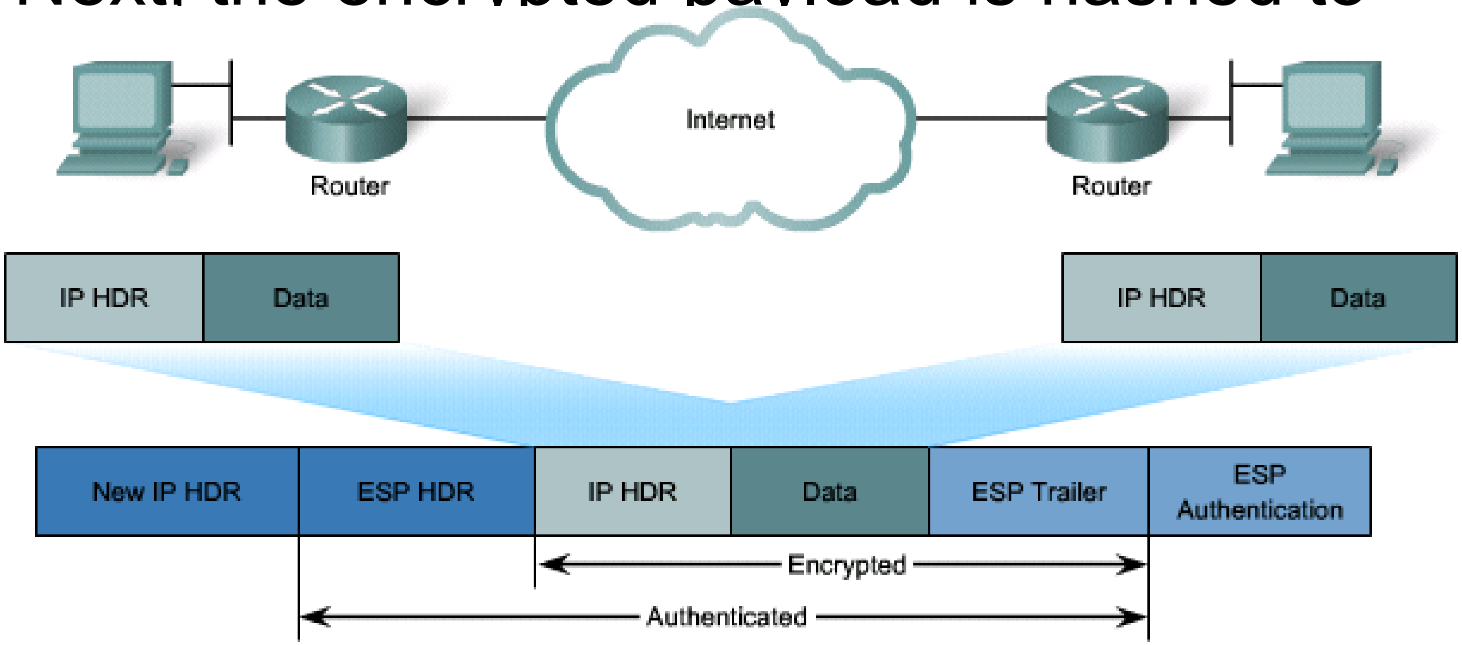
- It encapsulates the data to be protected.
- It operates on protocol number 50.



Encapsulating Security Payload (ESP)

- ESP can also provide integrity and authentication.
  - First, the payload is encrypted using DES (default), 3DES, AES, or SEAL.
  - Next, the encrypted payload is hashed to

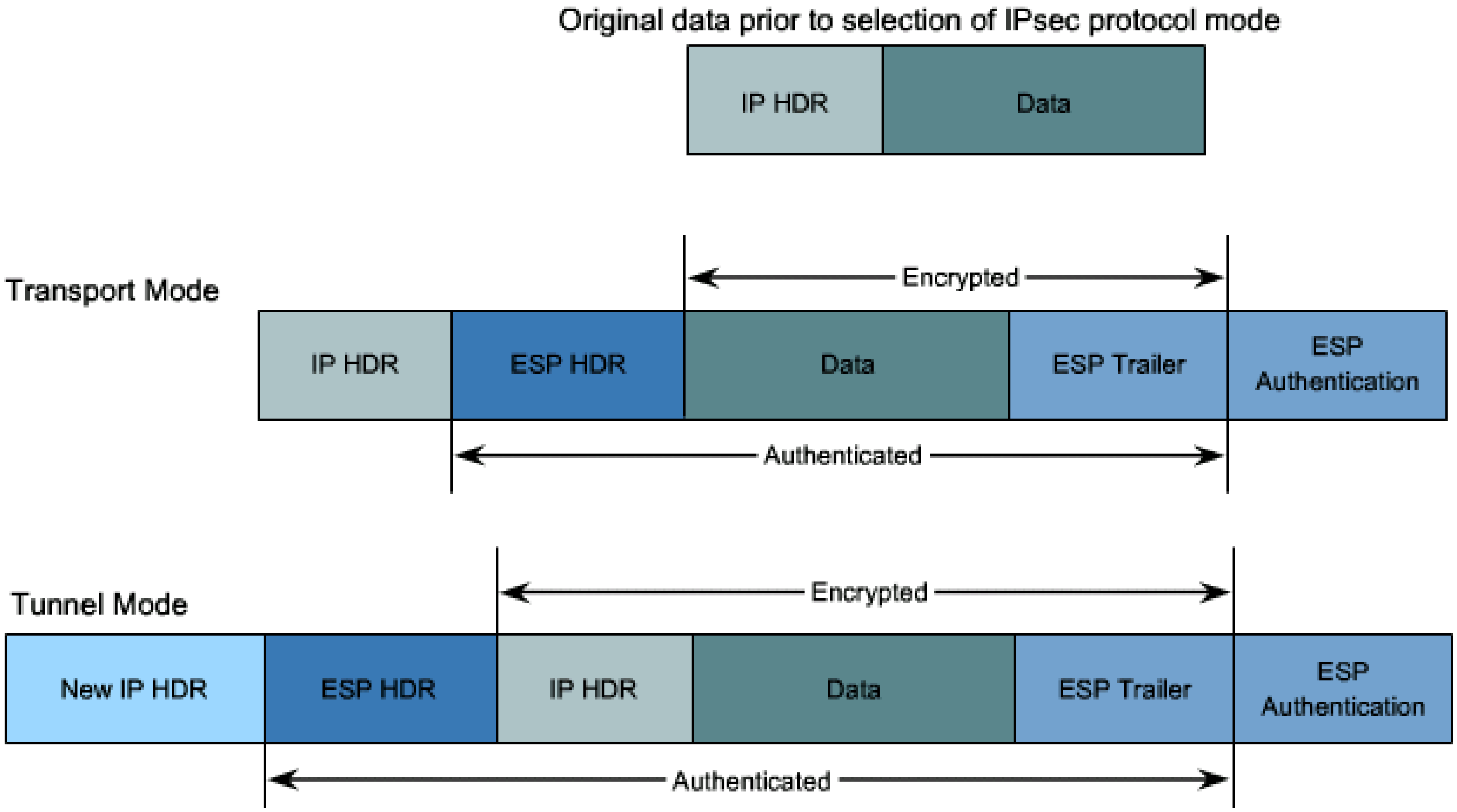
ng



- Provides confidentiality with encryption
- Provides integrity with authentication

Transport Mode and Tunnel Mode

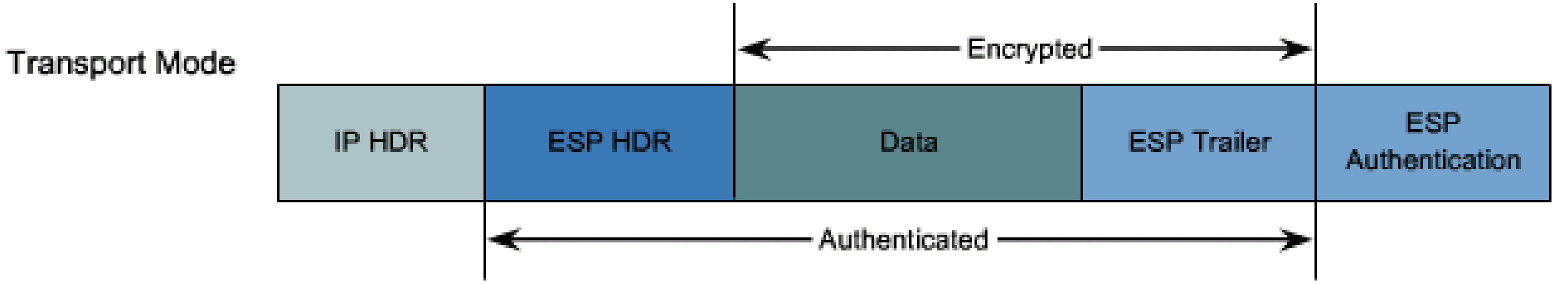
- ESP and AH can be applied to IP packets in two different modes.



Original data prior to selection of IPsec protocol mode

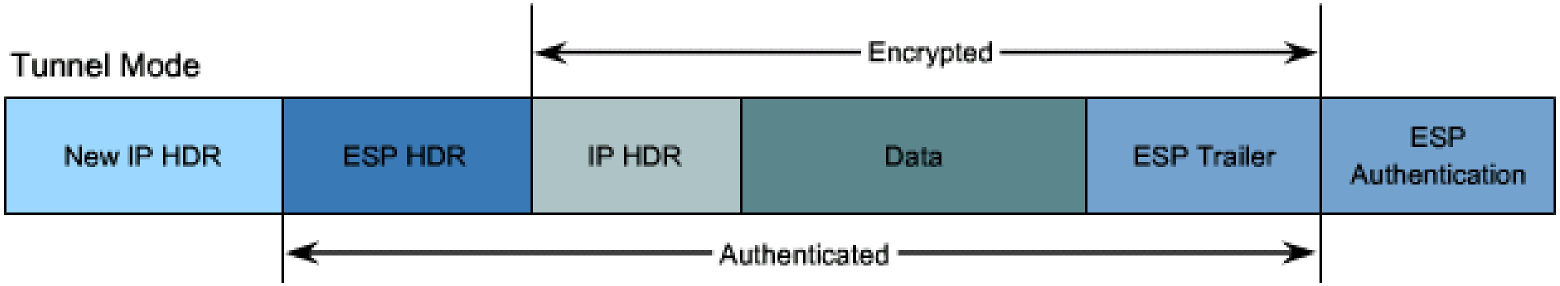
Transport Mode

- Security is provided only for the Transport Layer and above.
  - It protects the payload but leaves the original IP address in plaintext.
- ESP transport mode is used between hosts.
- Transport mode works well with GRE, because GRE hides the addresses of the end devices by adding its own IP.



Tunnel Mode

- Tunnel mode provides security for the complete original IP packet.
  - The original IP packet is encrypted and then it is encapsulated in another IP packet (IP-in-IP encryption).
- ESP tunnel mode is used in remote access and site-to-site implementations.



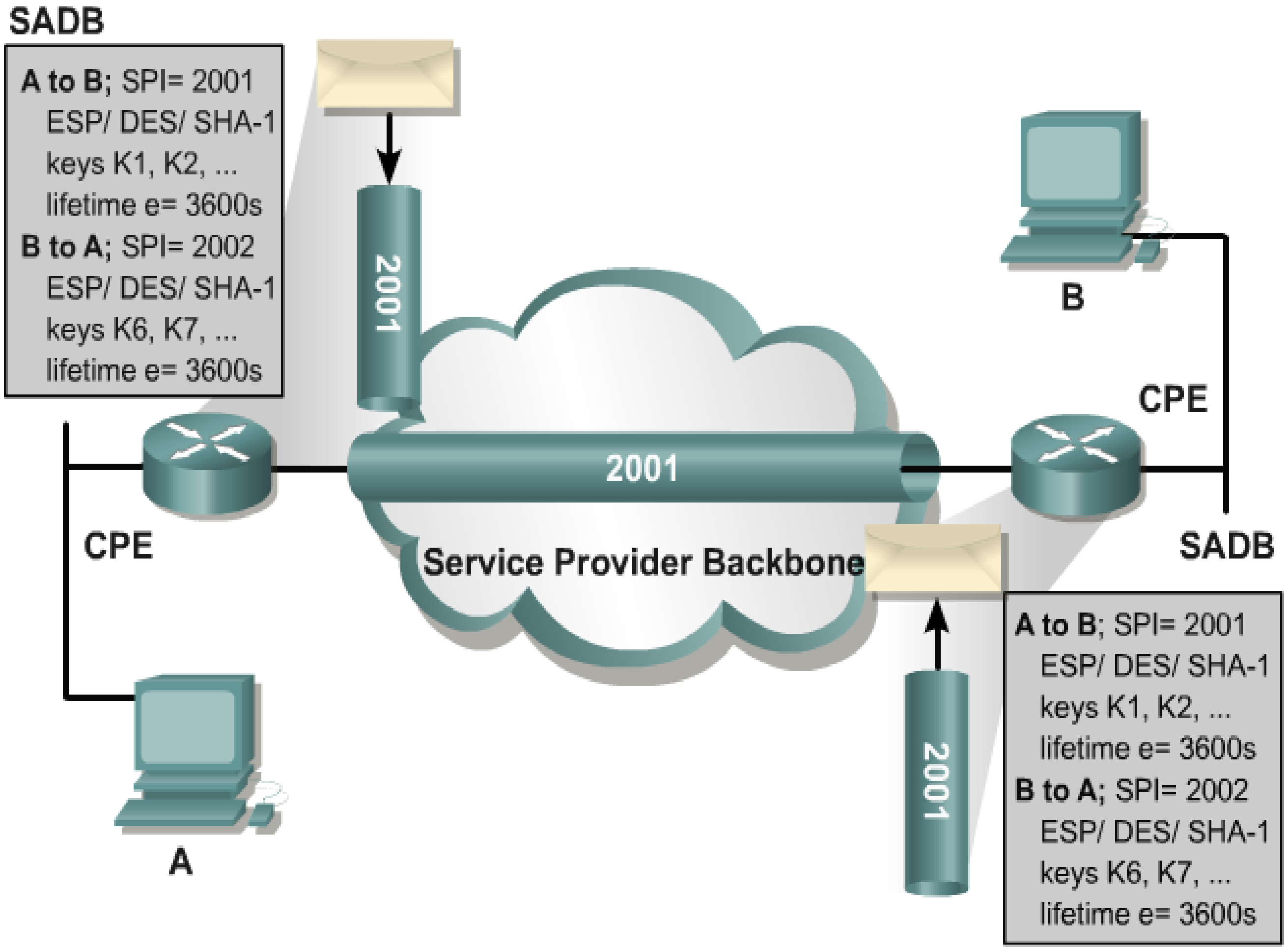
Key Exchange

- The IPsec VPN solution:
  - Negotiates key exchange parameters (IKE).
  - Establishes a shared key (DH).
  - Authenticates the peer.
  - Negotiates the encryption parameters.
- The negotiated parameters between two devices are known as a security association (SA).

Security Associations (SAs)

- 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 practically define the security policy used in IPsec.

SA Security Parameters



SADB

A to B; SPI= 2001
ESP/ DES/ SHA-1
keys K1, K2, ...
lifetime e= 3600s
B to A; SPI= 2002
ESP/ DES/ SHA-1
keys K6, K7, ...
lifetime e= 3600s

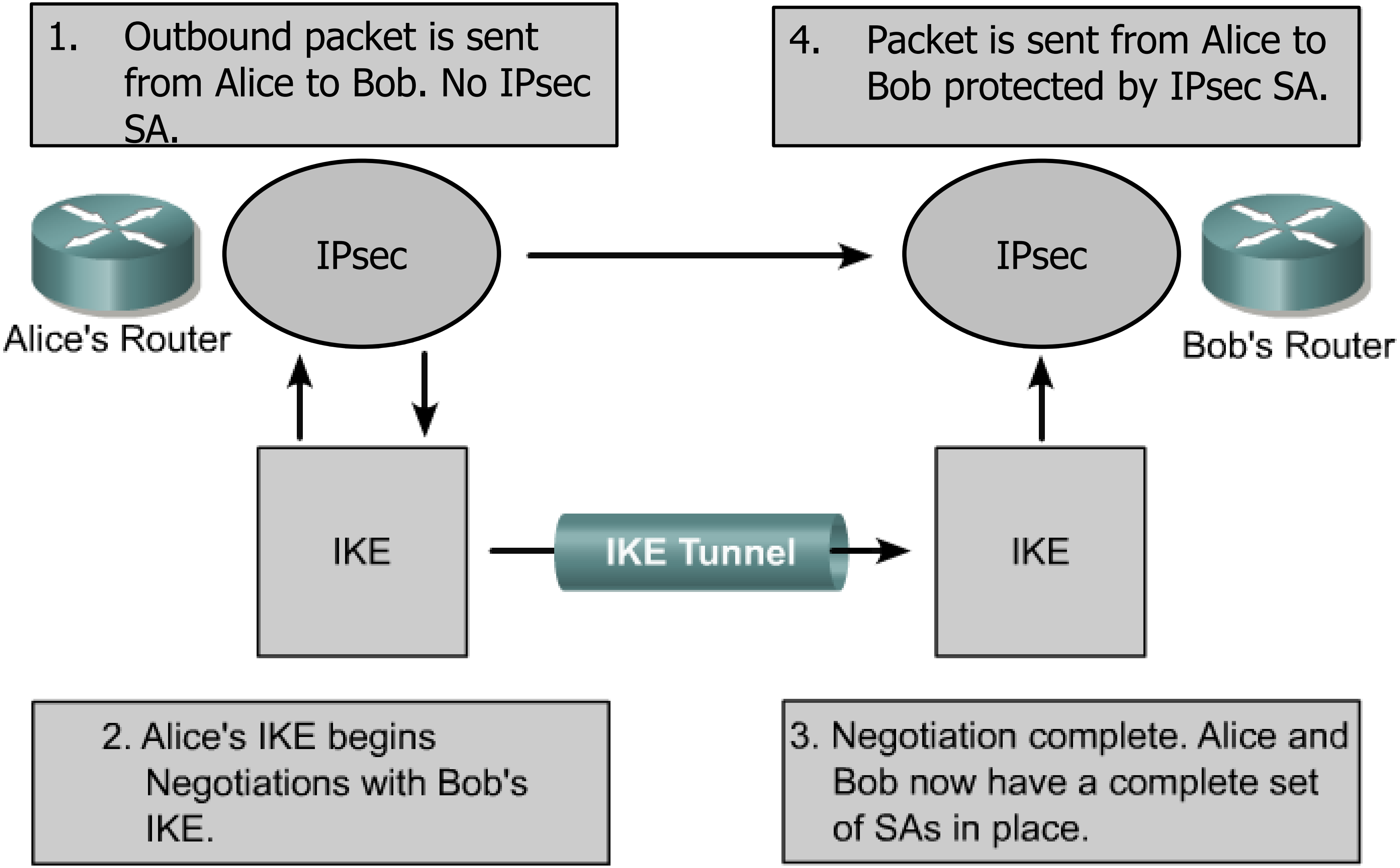
SADB

A to B; SPI= 2001
ESP/ DES/ SHA-1
keys K1, K2, ...
lifetime e= 3600s
B to A; SPI= 2002
ESP/ DES/ SHA-1
keys K6, K7, ...
lifetime e= 3600s

IKE - Internet Key Exchange

- IKE helps IPsec securely exchange cryptographic keys between distant devices.
  - Combination of the ISAKMP and the Oakley Key Exchange Protocol.
- Key Management can be preconfigured with IKE (ISAKMP) or with a manual key configuration.
  - IKE and ISAKMP are often used interchangeably.
- The IKE tunnel protects the SA negotiations.
  - After the SAs are in place, IPsec protects the data that Alice and Bob exchange.

How IPsec uses IKE



1. Outbound packet is sent from Alice to Bob. No IPsec SA.

4. Packet is sent from Alice to Bob protected by IPsec SA.

2. Alice's IKE begins Negotiations with Bob's IKE.

3. Negotiation complete. Alice and Bob now have a complete set of SAs in place.

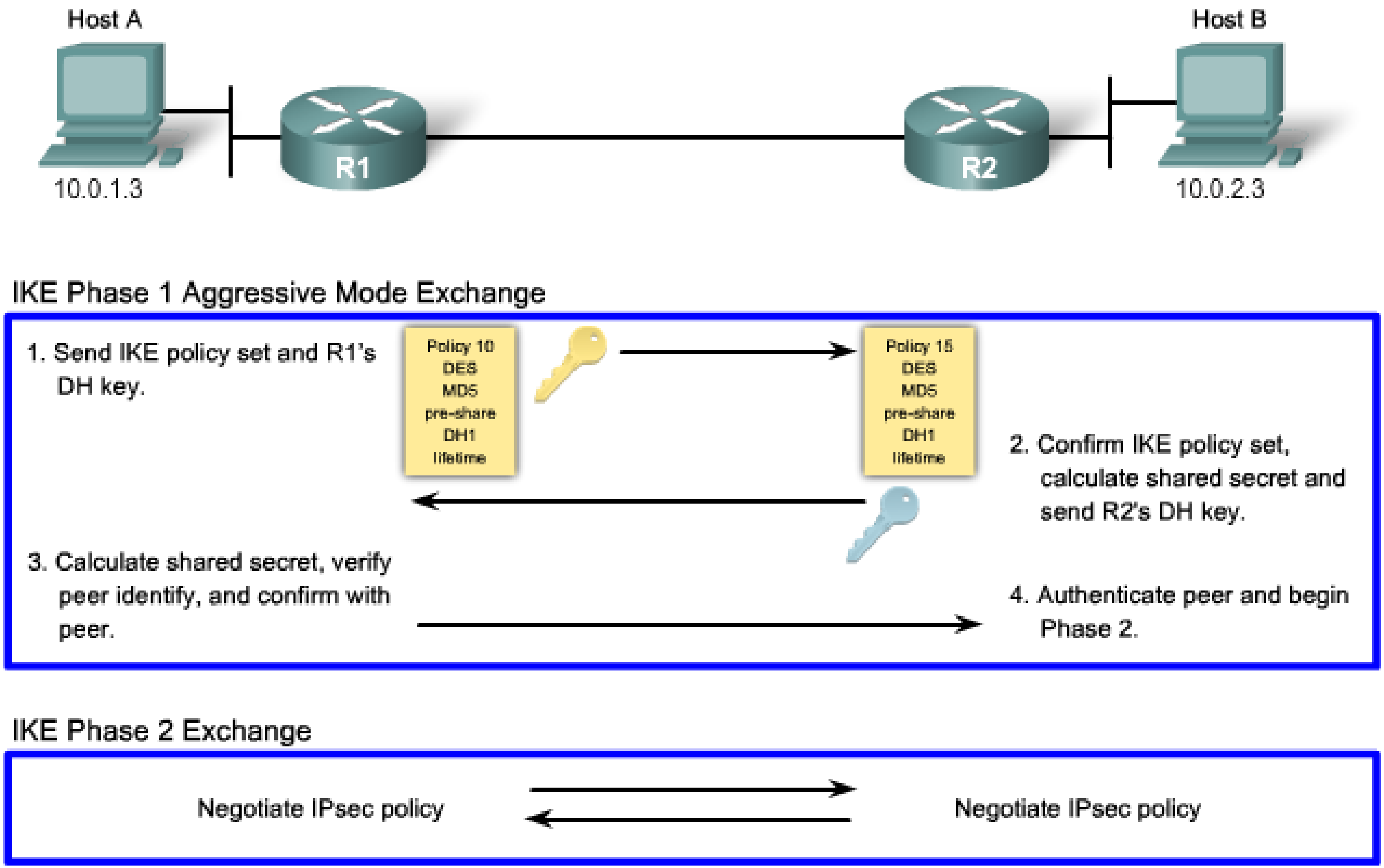
IKE - Internet Key Exchange

- There are two phases in every IKE negotiation
    - Phase 1 (Authentication)
    - Phase 2 (Key Exchange)
- IKE negotiation can also occur in:
    - Main Mode
    - Aggressive mode
- The difference between the two is that Main mode requires the exchange of 6 messages while Aggressive mode requires only 3 exchanges.

IKE Main Mode Phases

- IKE Phase One:
  - Negotiates an IKE protection suite.
  - Exchanges keying material to protect the IKE session (DH).
  - Authenticates each other.
  - Establishes the IKE SA.
  - Main Mode requires the exchange of 6 messages while Aggressive mode only uses 3 messages.
- IKE Phase Two:
  - Negotiates IPsec security parameters, known as IPsec transform sets.
  - Establishes IPsec SAs.
  - Periodically renegotiates IPsec SAs to ensure security.
  - Optionally performs an additional DH exchange.

IKE Phases



IKE Phase 1 Aggressive Mode Exchange

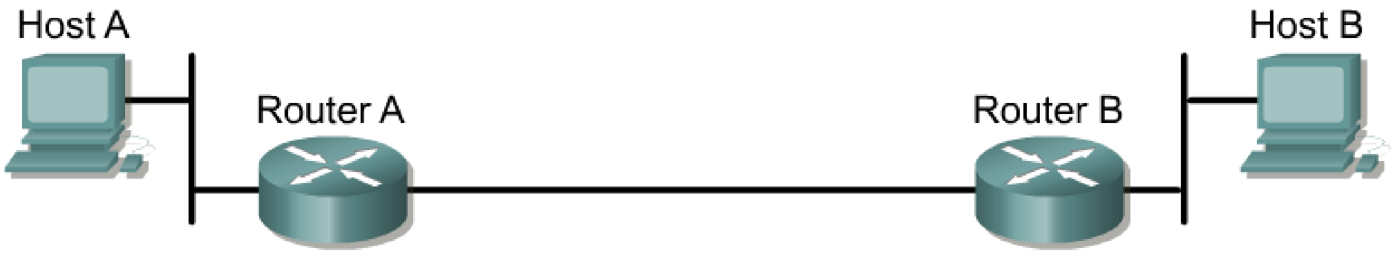
1. Send IKE policy set and R1's DH key.
2. Confirm IKE policy set, calculate shared secret and send R2's DH key.
3. Calculate shared secret, verify peer identify, and confirm with peer.
4. Authenticate peer and begin Phase 2.

IKE Phase 2 Exchange

Negotiate IPsec policy

Negotiate IPsec policy

Five Steps of IPsec



Step 1 Host A sends interesting traffic destined for Host B.

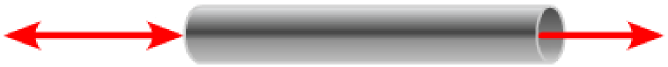
Step 2 **IKE Phase 1** authenticates IPsec peers and negotiates IKE SAs to create a secure communications channel for negotiating IPsec SAs in Phase 2.

IKE SA ⟷ IKE Phase 1 ⟷ IKE SA

Step 3 **IKE Phase 2** negotiates IPsec SA parameters and creates matching IPsec SAs in the peers to protect data and messages exchanged between endpoints.

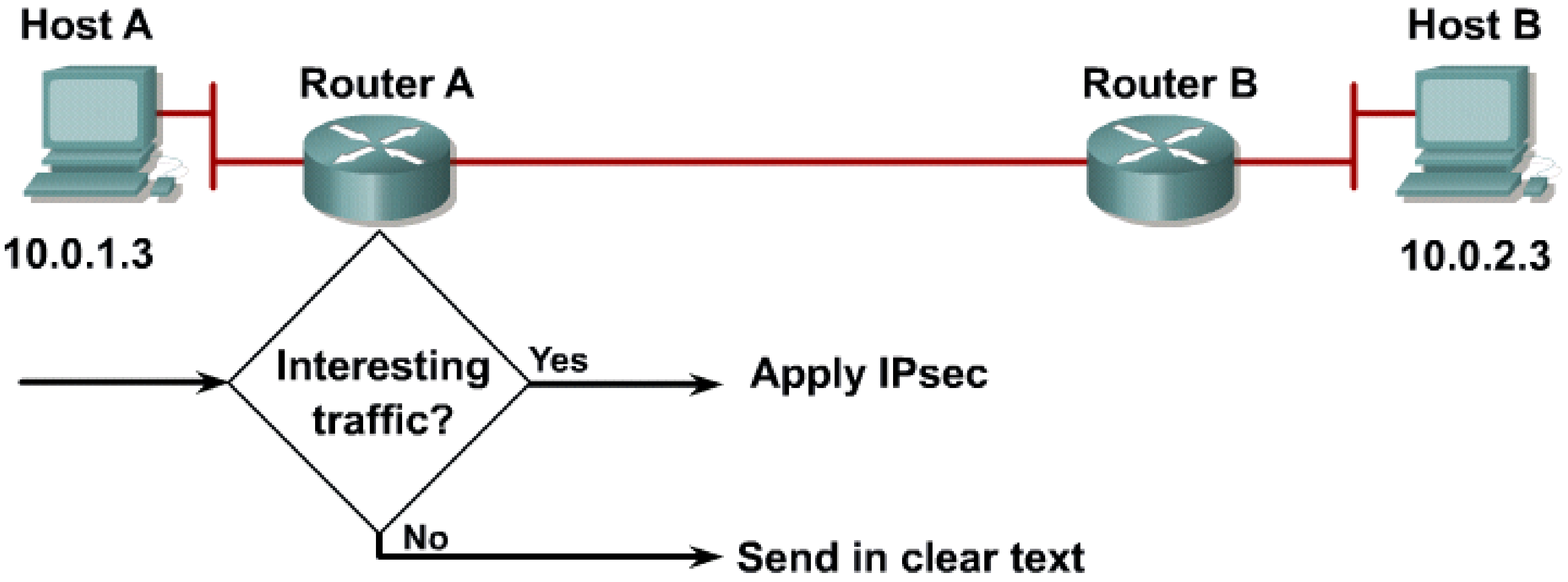
IKE SA ⟷ IKE Phase 2 ⟷ IKE SA

Step 4 Data transfer occurs between IPsec peers based on the IPsec parameters and keys stored in the SA database.



Step 5 IPsec tunnel termination occurs by SAs through deletion or by timing out.

Step 1 – Interesting Traffic



Host A

Router A

Router B

Host B

10.0.1.3

10.0.2.3

Interesting traffic?

Yes

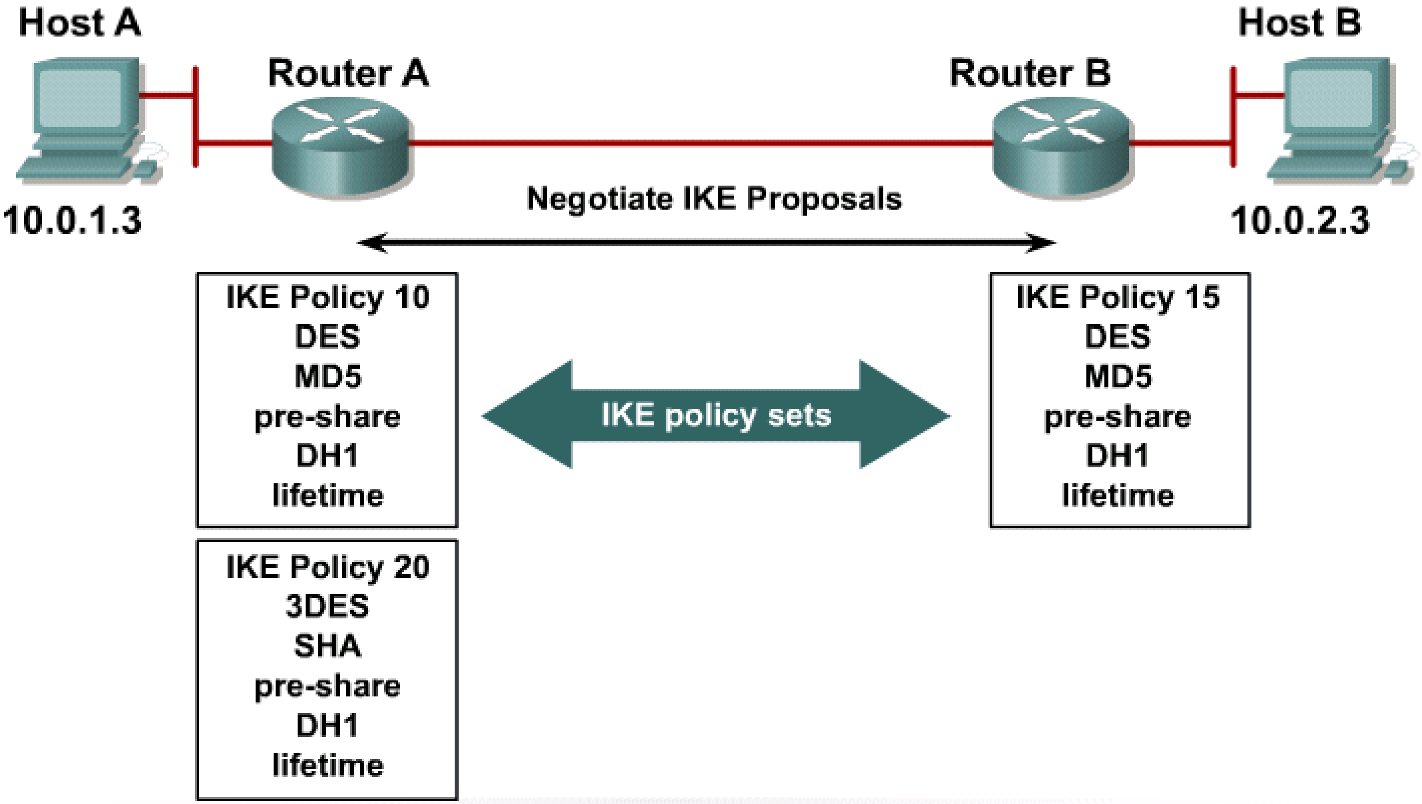
Apply IPsec

No

Send in clear text

Step 2 – IKE Phase 1

IKE Policy Negotiation



IKE Policy 10
DES
MD5
pre-share
DH1
lifetime

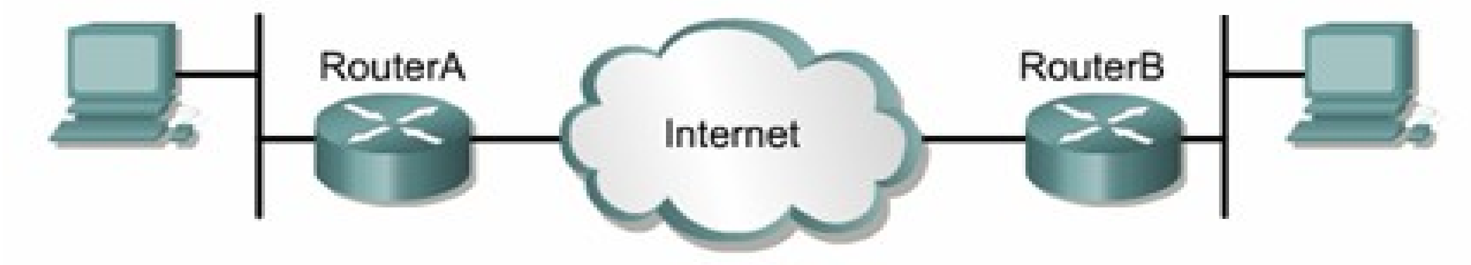
IKE Policy 20
3DES
SHA
pre-share
DH1
lifetime

IKE Policy 15
DES
MD5
pre-share
DH1
lifetime

Negotiates matching IKE transform sets to protect IKE exchange

Step 2 – IKE Phase 1

DH Key Exchange



RouterA randomly chooses a string and sends it to RouterB.

RouterB hashes the received string together with the pre-shared secret and yields a hash value.

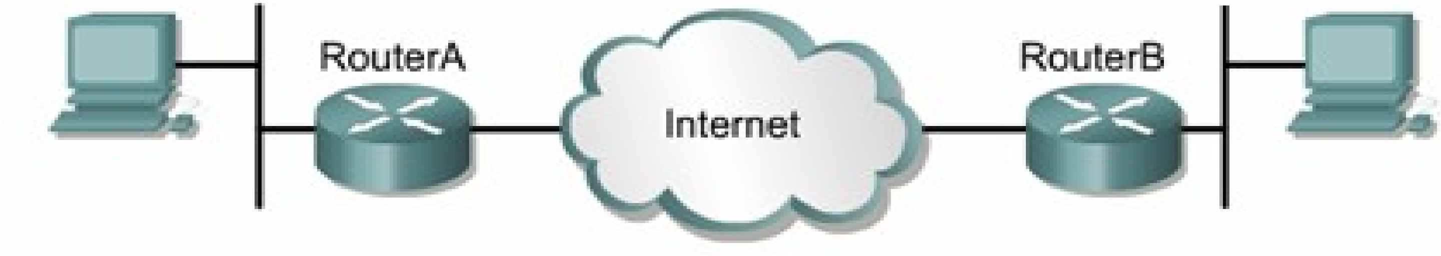
RouterB sends the result of hashing back to RouterA.

RouterA calculates its own hash of the random string, together with the pre-shared secret, and matches it with the received result from the other peer.

If they match, RouterB knows the pre-shared secret, and is considered authenticated.

Step 2 – IKE Phase 1

DH Key Exchange



Now RouterB randomly chooses a different random string and sends it to RouterA.

RouterA also hashes the received string together with the pre-shared secret and yields a hash value.

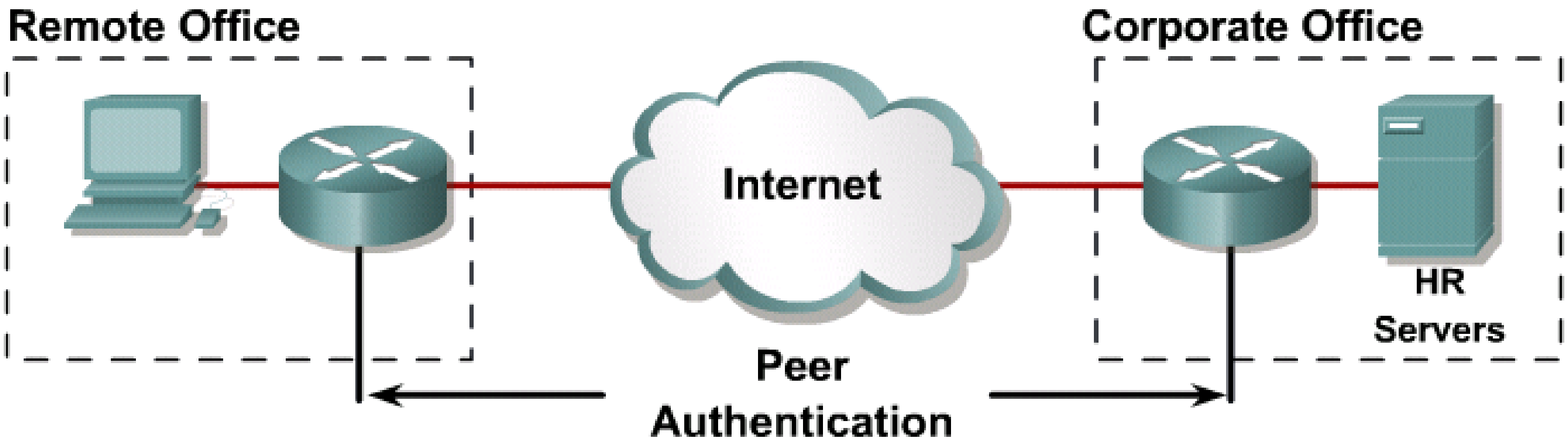
RouterA sends the result of hashing back to RouterB.

RouterB calculates its own hash of the random string, together with the pre-shared secret, and matches it with the received result from the other peer.

If they match, RouterA knows the pre-shared secret, and is considered authenticated.

Step 2 – IKE Phase 1

Peer Authentication

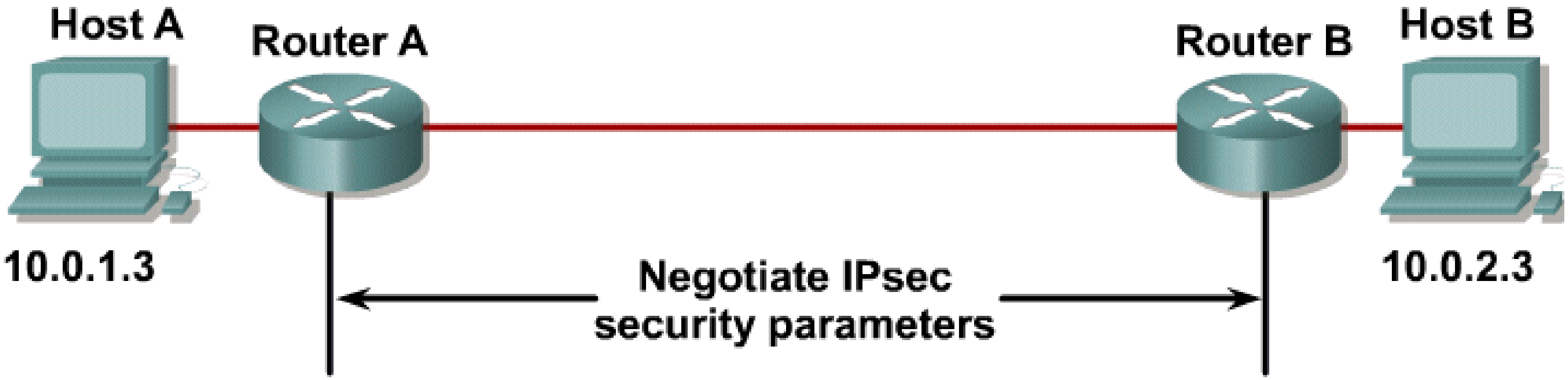


Peer authentication methods:

- Pre-shared keys
- RSA signatures
- RSA encrypted nonces

Step 3 – IKE Phase 2

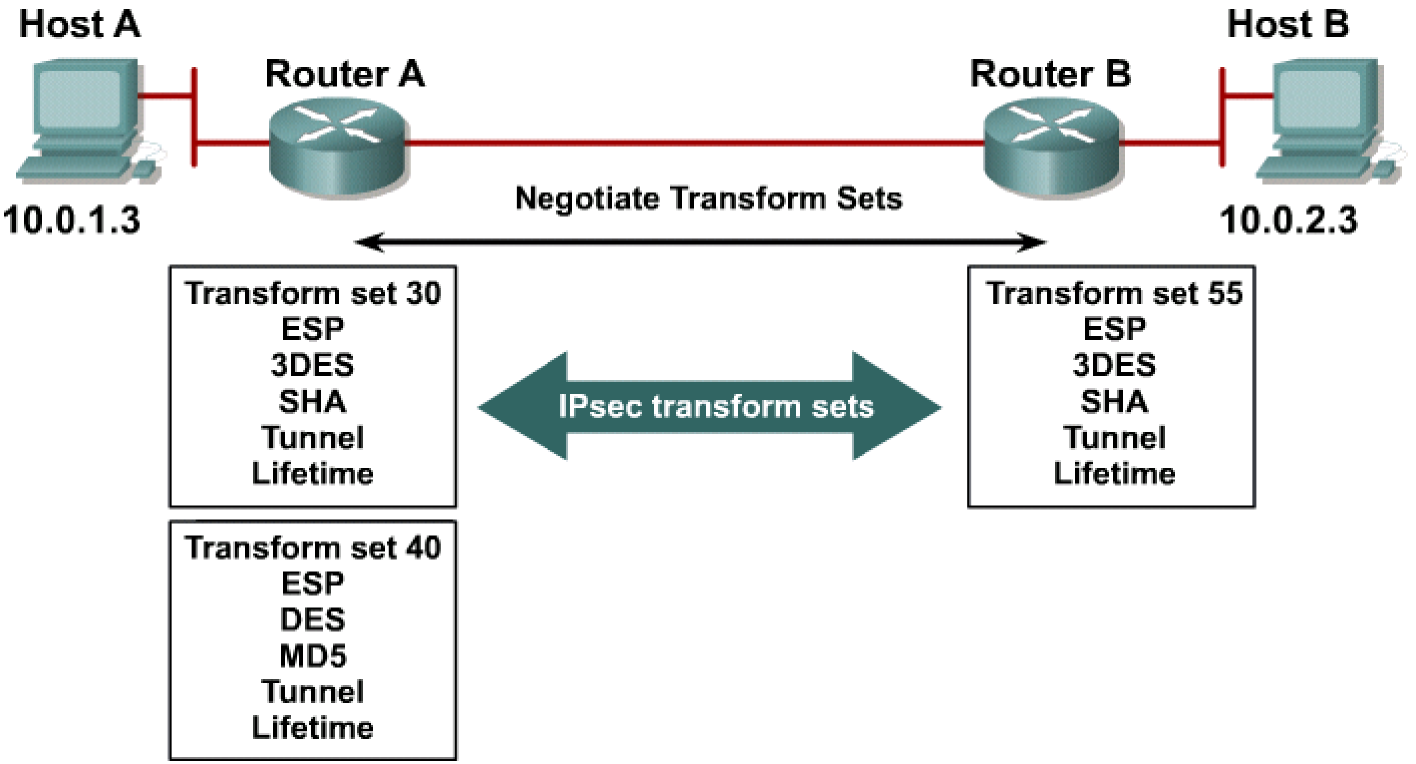
IPsec Negotiation



- Negotiates IPsec security parameters and IPsec transform sets
- Establishes IPsec SAs
- Periodically renegotiates IPsec SAs to ensure security
- Optionally, performs an additional Diffie-Hellman exchange

Step 3 – IKE Phase 2

Transform Set Negotiation

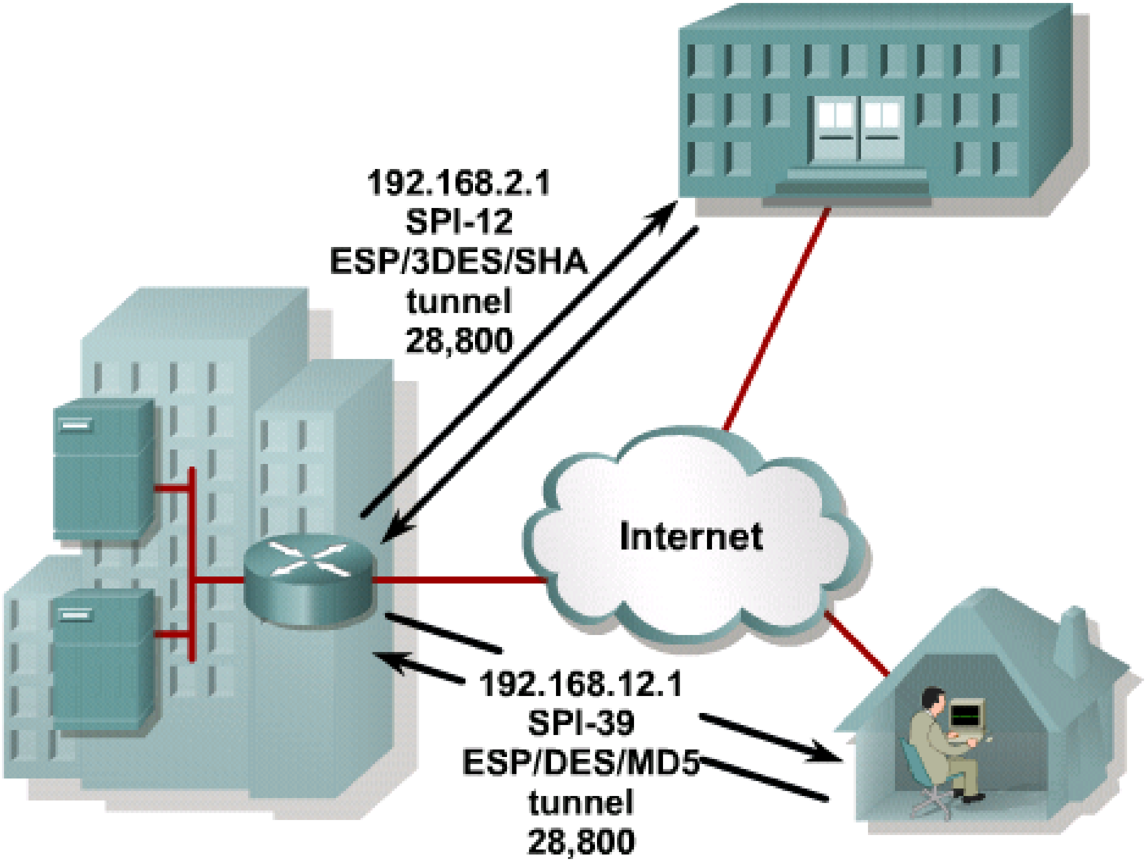


A transform set is a combination of algorithms and protocols that enact a security policy for traffic.

Step 3 – IKE Phase 2

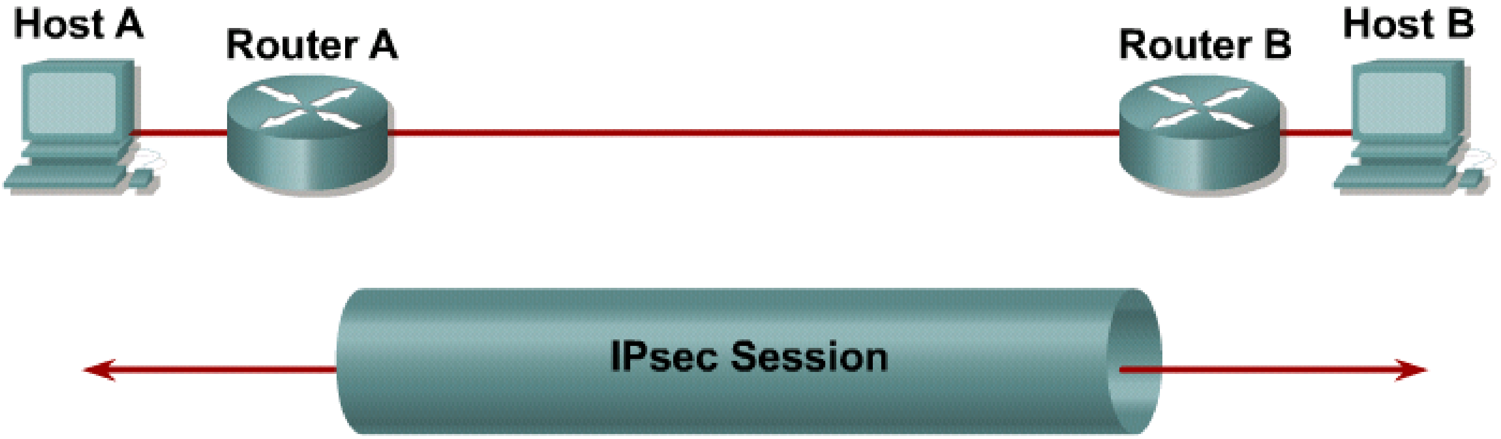
Security Associations

- SA database:
  - Destination IP address
  - SPI
  - Protocol (ESP or AH)
- Security policy database:
  - Encryption algorithm
  - Authentication algorithm
  - Mode
  - Key lifetime



Step 4

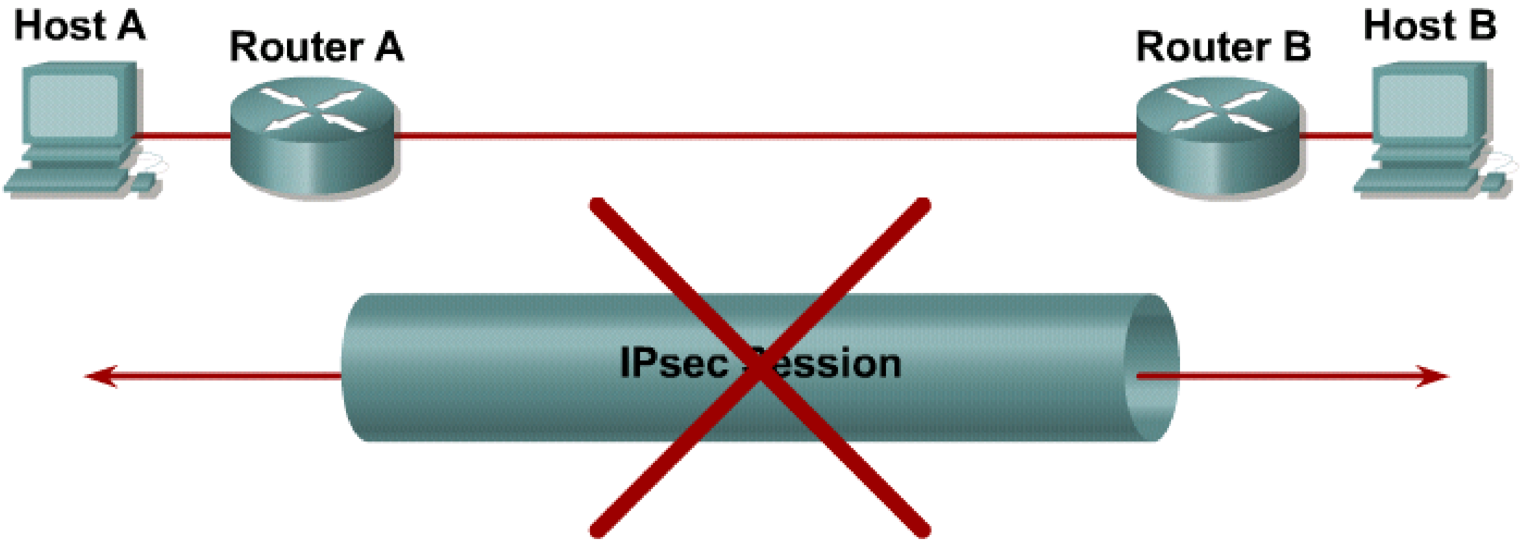
IPsec Session



- SAs are exchanged between peers.
- The negotiated security services are applied to the traffic.

Step 5

Tunnel Termination



- A tunnel is terminated by one of the following:
  - By an SA lifetime timeout
  - The packet counter is exceeded
- IPsec SA is removed.

Configuring IPsec

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.

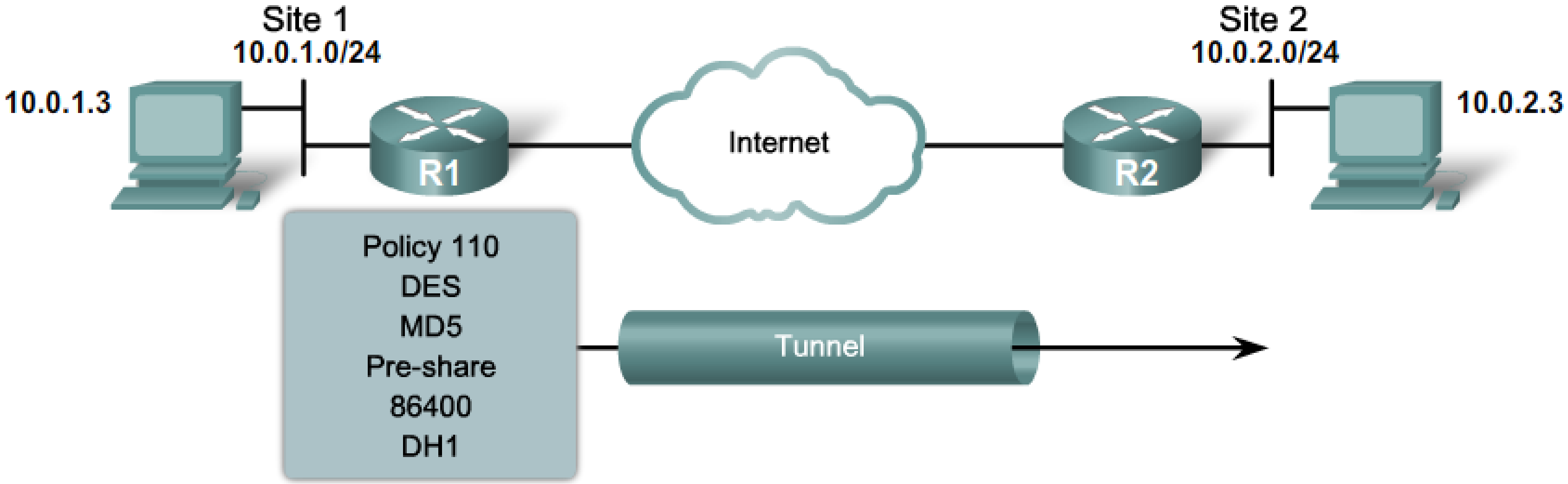
IPsec Tasks

1. Ensure that ACLs configured on the interface are compatible with IPsec configuration.
2. Create an IKE policy to determine the parameters that will be used to establish the tunnel.
3. Configure the IPsec transform set which defines the parameters that the IPsec tunnel uses.
    - The set can include the encryption and integrity algorithms.
4. Create a crypto ACL.
    - The crypto ACL defines which traffic is sent through the IPsec tunnel and protected by the IPsec process.
5. Create and apply a crypto map.
    - The crypto map groups the previously configured parameters together and defines the IPsec peer devices.
    - The crypto map is applied to the outgoing interface of the VPN device.

Default ISAKMP Settings

```
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
```

Create an IKE Policy



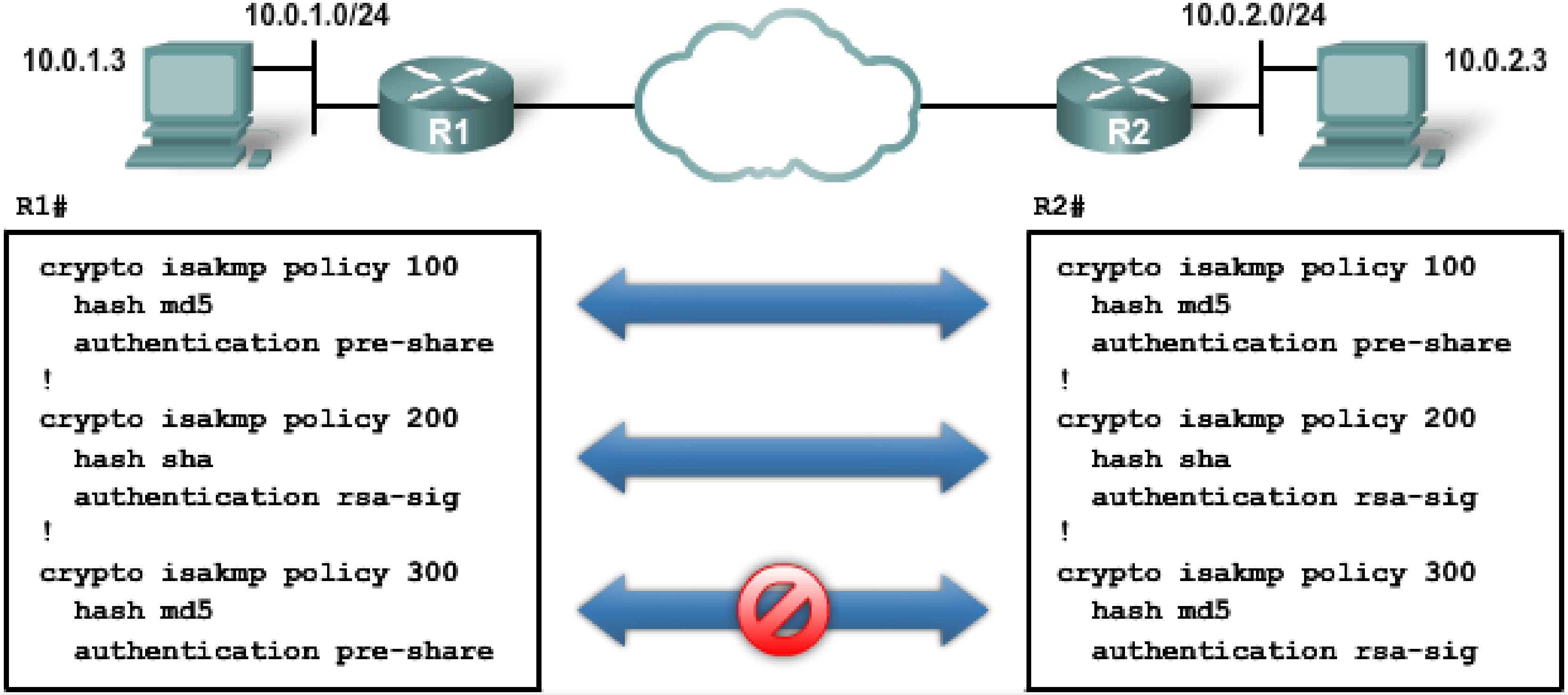
```
router(config)#
```

```
crypto isakmp policy priority
```

Defines the parameters within the IKE policy

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

ISAKMP Policy Negotiation



R1#

```
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
```

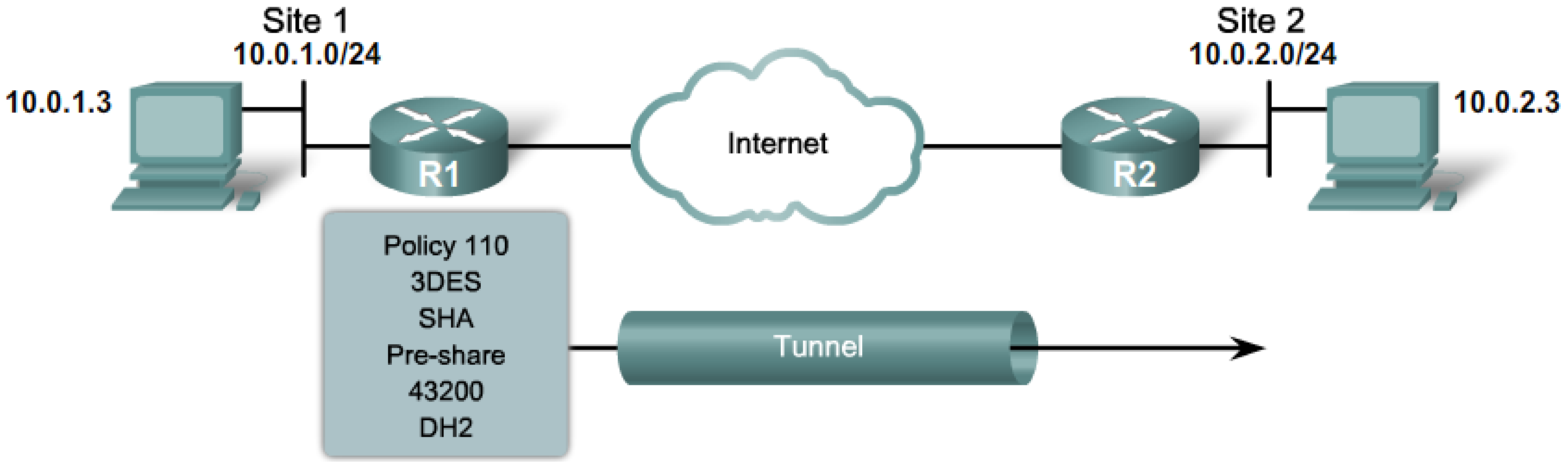
R2#

```
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
```

Policy 100 and 200 can be successfully negotiated, but policy 300 cannot.

Note: smaller priority numbers have higher priority.

ISAKMP Policy Negotiation

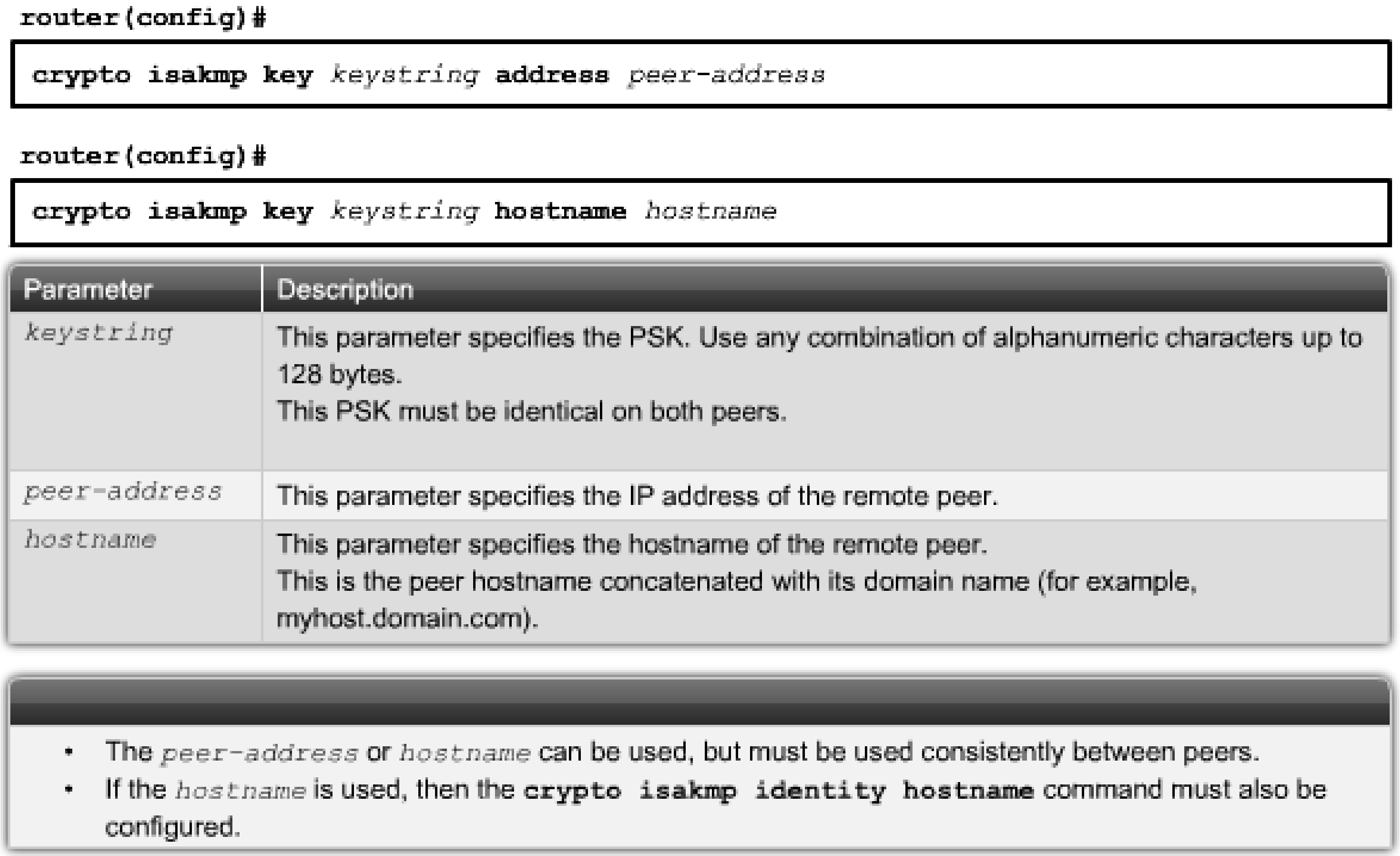


```
R1(config)# crypto isakmp policy 110
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# encryption 3des
R1(config-isakmp)# group 2
R1(config-isakmp)# hash sha
R1(config-isakmp)# lifetime 43200
```

```
R2(config)# crypto isakmp policy 100
R2(config-isakmp)# authentication pre-share
R2(config-isakmp)# encryption 3des
R2(config-isakmp)# group 2
R2(config-isakmp)# hash sha
R2(config-isakmp)# lifetime 43200
```

- R1 attempts to establish a VPN tunnel with R2 because it has interesting traffic destined for R2 and therefore sends its IKE policy parameters.
- R2 must have an ISAKMP policy configured with the same parameters. Notice however, that policy numbers are only locally significant and do not have to match between IPsec peers.

Configure Pre-Shared Keys



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

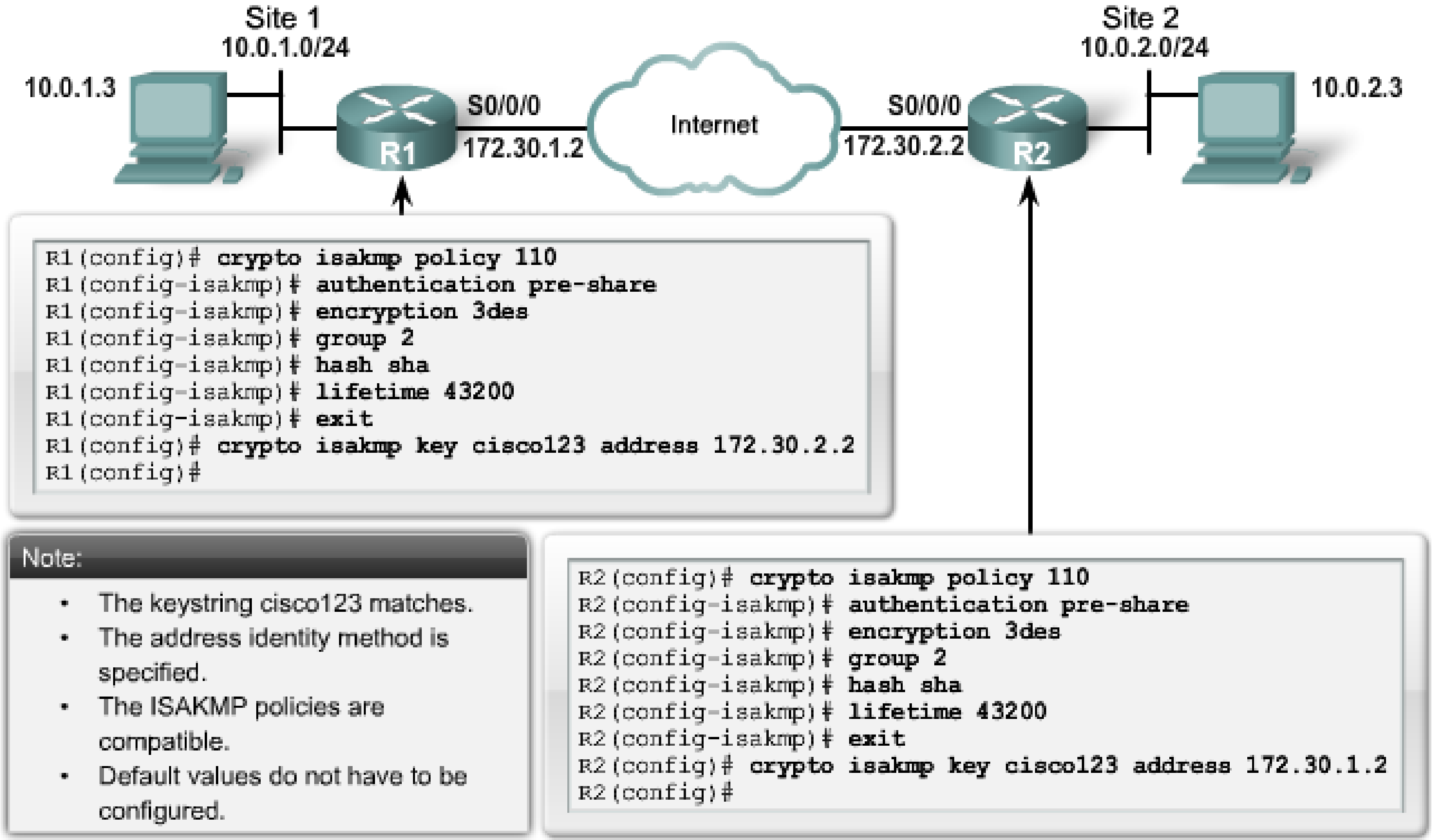
```
router(config)#
crypto isakmp key keystring hostname hostname
```

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).

- The *peer-address* or *hostname* can be used, but must be used consistently between peers.
- If the *hostname* is used, then the `crypto isakmp identity hostname` command must also be configured.

- By default, the ISAKMP identity is set to use the IP address.

Configure Pre-Shared Keys



```
R1(config)# crypto isakmp policy 110
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# encryption 3des
R1(config-isakmp)# group 2
R1(config-isakmp)# hash sha
R1(config-isakmp)# lifetime 43200
R1(config-isakmp)# exit
R1(config)# crypto isakmp key cisco123 address 172.30.2.2
R1(config)#
```

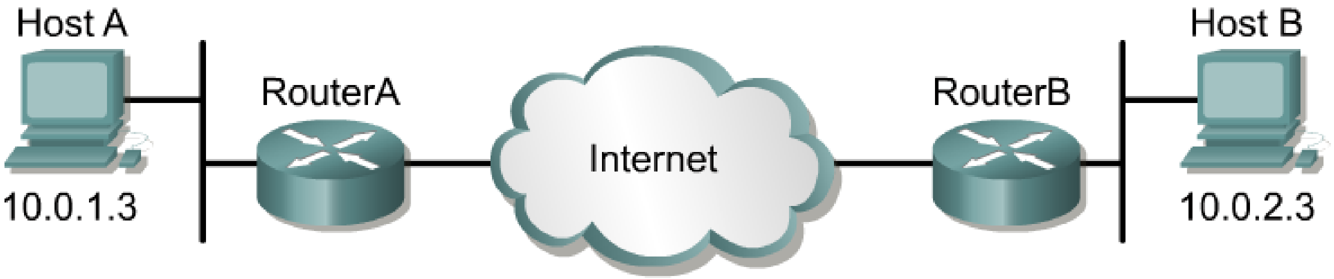
Note:

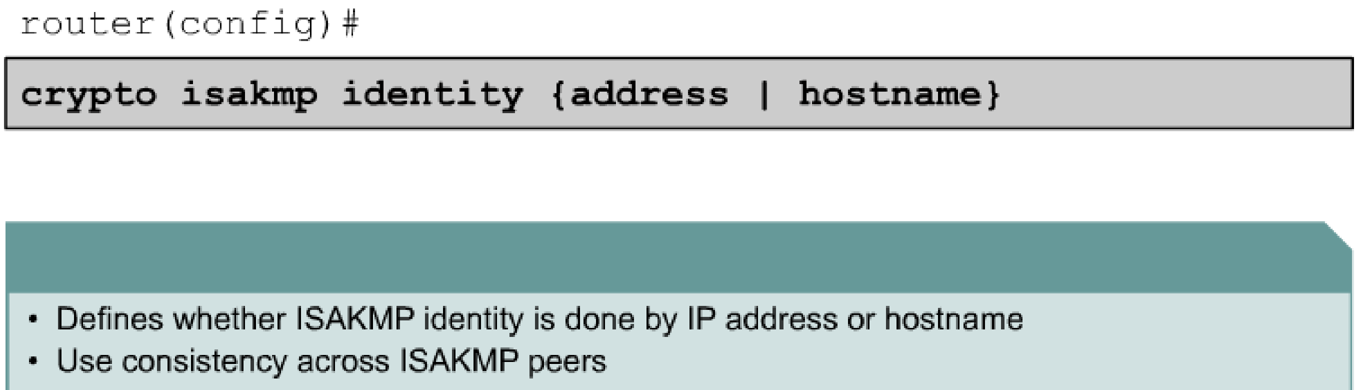
- The keystring cisco123 matches.
- The address identity method is specified.
- The ISAKMP policies are compatible.
- Default values do not have to be configured.

```
R2(config)# crypto isakmp policy 110
R2(config-isakmp)# authentication pre-share
R2(config-isakmp)# encryption 3des
R2(config-isakmp)# group 2
R2(config-isakmp)# hash sha
R2(config-isakmp)# lifetime 43200
R2(config-isakmp)# exit
R2(config)# crypto isakmp key cisco123 address 172.30.1.2
R2(config)#
```

Configure ISAKMP Identity

- To use the **`hostname`** parameter, configure the **`crypto isakmp identity`** *`hostname`* global configuration mode command.
  - In addition, DNS must be accessible to resolve the hostname.

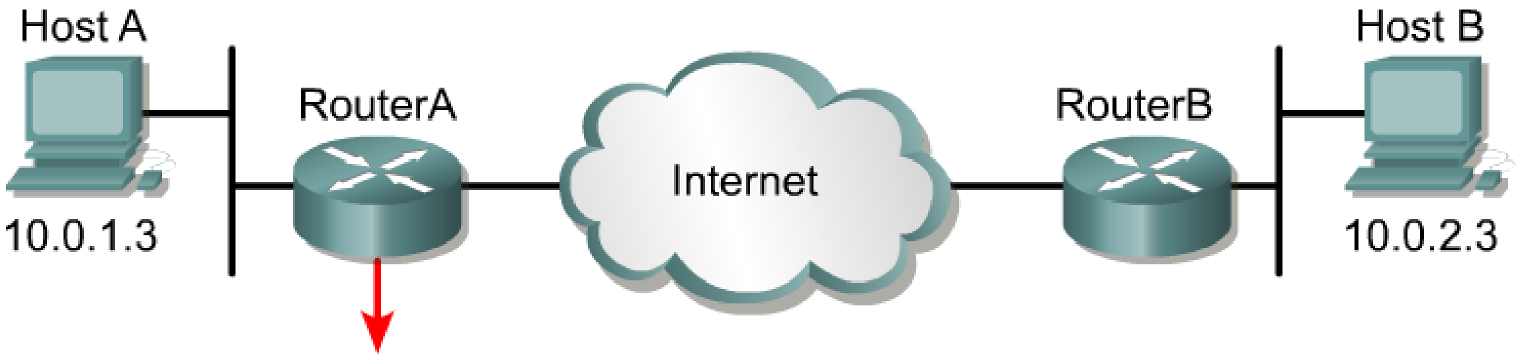




```
router(config)#
crypto isakmp identity {address | hostname}
```

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

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 the Transform Sets

- Determine the following policy details:
  - IPsec algorithms and parameters for optimal security and performance
  - Transforms sets
  - IPsec peer details
  - IP address and applications of hosts to be protected
  - Manual or IKE-initiated SAs
- Goal: Minimize misconfiguration.

IPsec Transforms Supported in IOS

- Cisco IOS software supports the following IPsec transforms:

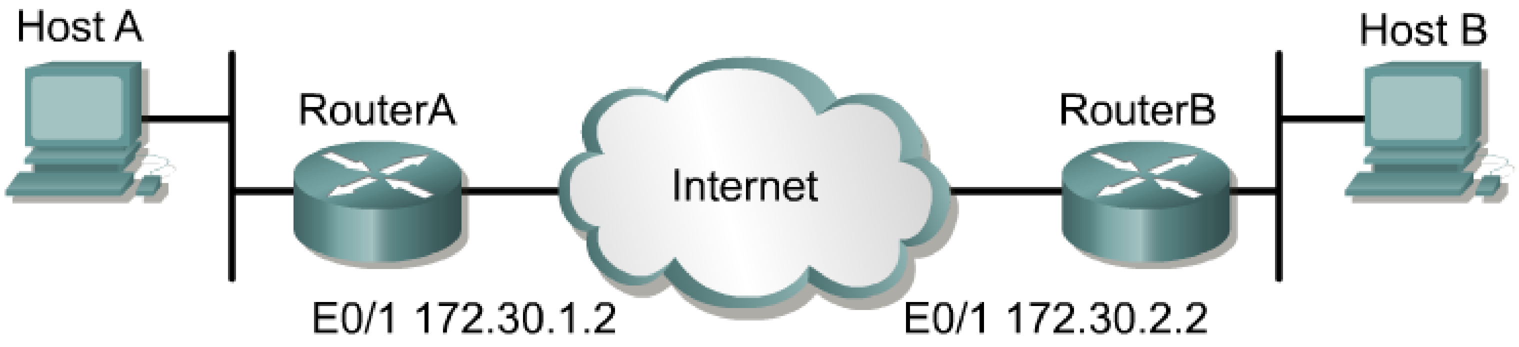
```
CentralA(config)# crypto ipsec transform-set transform-set-name ?
ah-md5-hmac   AH-HMAC-MD5 transform
ah-sha-hmac   AH-HMAC-SHA transform
esp-3des      ESP transform using 3DES(EDE) cipher (168 bits)
esp-des       ESP transform using DES cipher (56 bits)
esp-md5-hmac  ESP transform using HMAC-MD5 auth
esp-sha-hmac  ESP transform using HMAC-SHA auth
esp-null      ESP transform w/o cipher
```

Note:

esp-md5-hmac and **esp-sha-hmac** provide more data integrity.

They are compatible with NAT/PAT and are used more frequently than **ah-md5-hmac** and **ah-sha-hmac**.

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

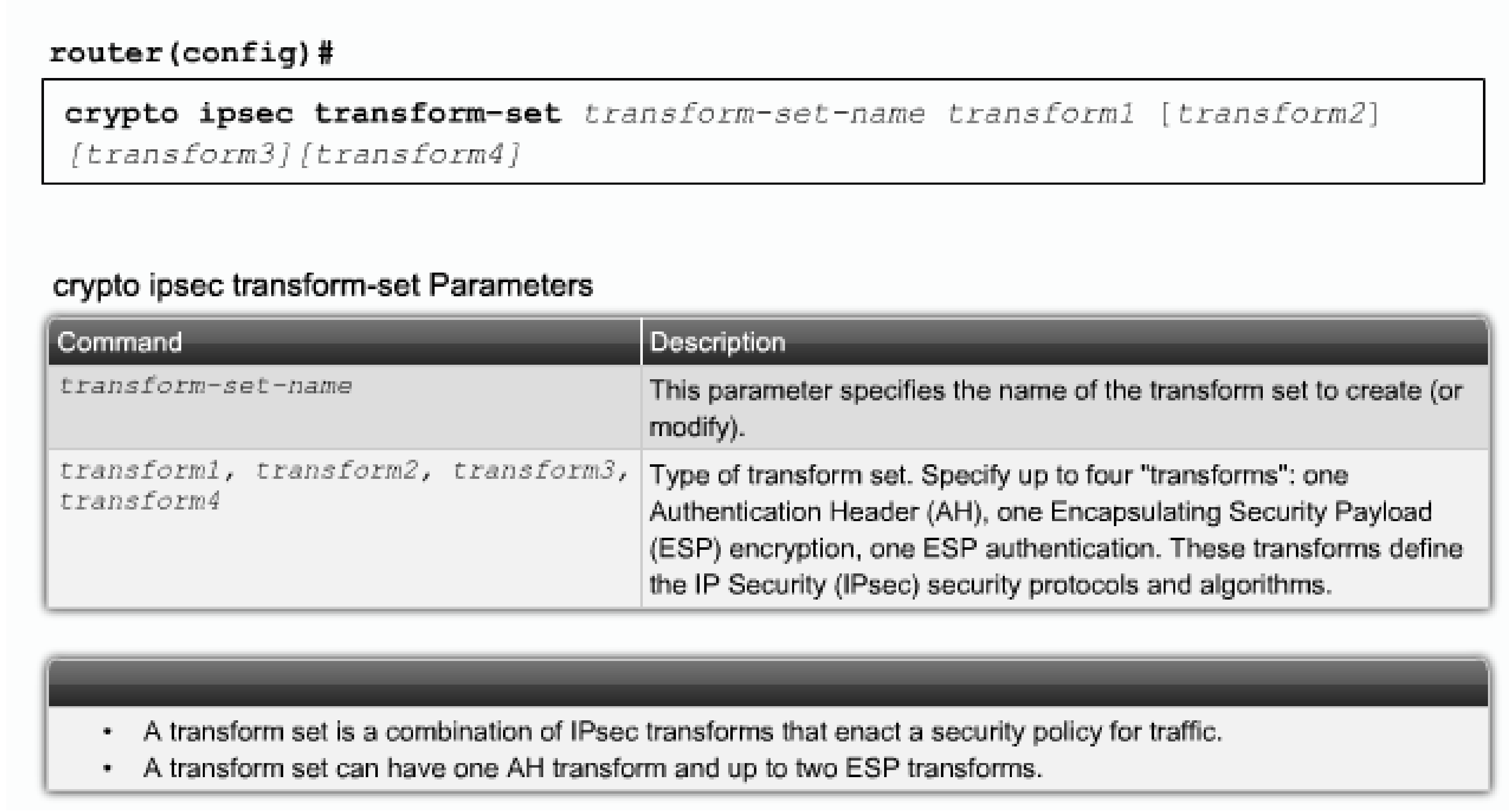
Specific IPsec `show` Commands

```
RouterA# show crypto isakmp policy
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
```

```
RouterA# show crypto map
Crypto Map “MYMAP" 10 ipsec-isakmp
Peer = 172.30.2.2
Extended IP access list 102
access-list 102 permit ip host 172.30.1.2 host 172.30.2.2
Current peer: 172.30.2.2
Security association lifetime: 4608000 kilobytes/3600 seconds
PFS (Y/N): N
Transform sets={ MY-SET, }
```

```
RouterA# show crypto ipsec transform-set MY-SET
Transform set MY-SET: { esp-des }
will negotiate = { Tunnel, },
```

Configure Transform Sets



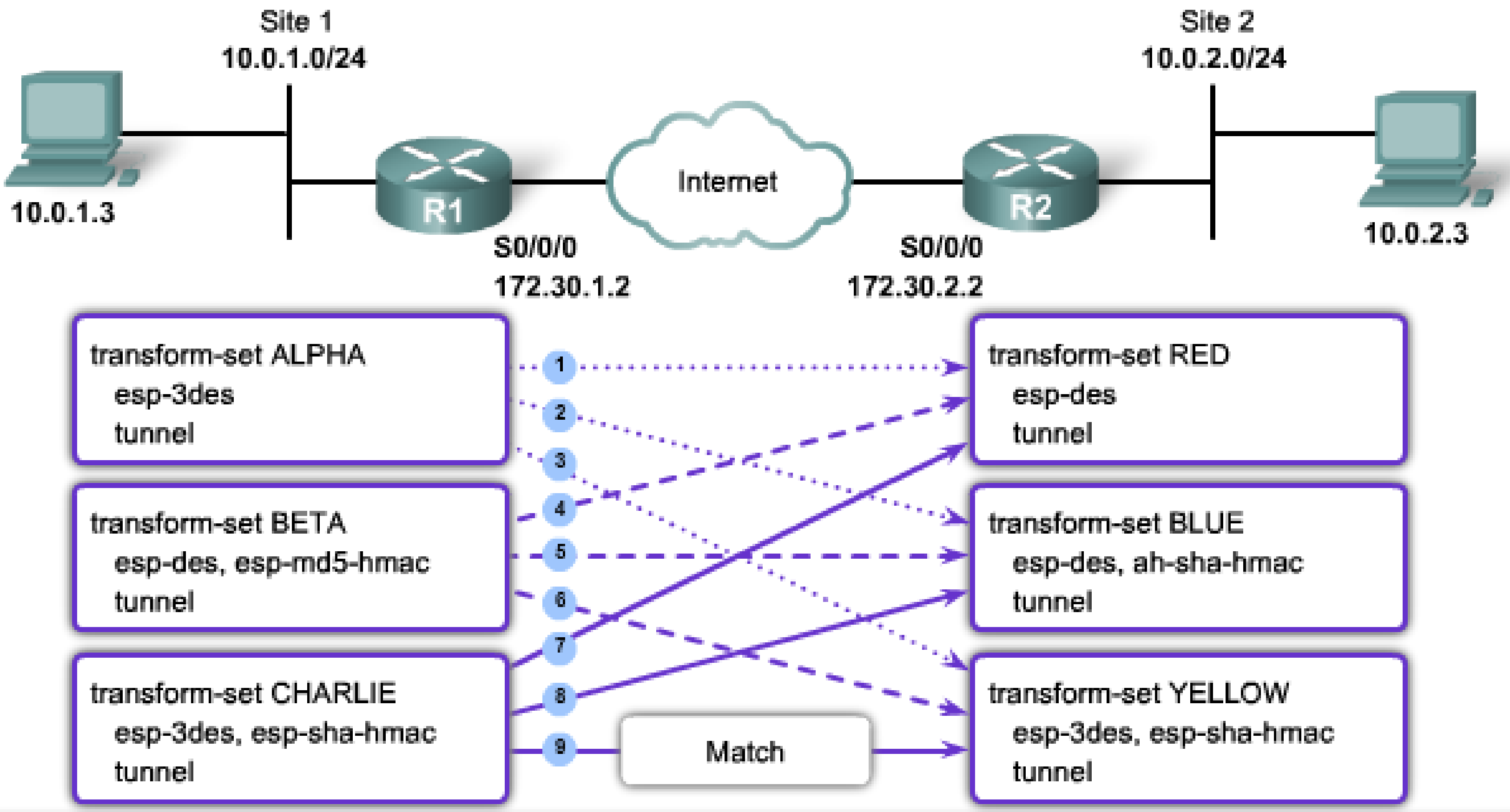
```
router(config)#
crypto ipsec transform-set transform-set-name transform1 [transform2]
[transform3][transform4]
```

crypto ipsec transform-set Parameters

Command	Description
transform-set-name	This parameter specifies the name of the transform set to create (or modify).
transform1, transform2, transform3, transform4	Type of transform set. Specify up to four "transforms": one Authentication Header (AH), one Encapsulating Security Payload (ESP) encryption, one ESP authentication. These transforms define the IP Security (IPsec) security protocols and algorithms.

- A transform set is a combination of IPsec transforms that enact a security policy for traffic.
- A transform set can have one AH transform and up to two ESP transforms.

Transform Set Negotiation



transform-set ALPHA
esp-3des
tunnel

transform-set BETA
esp-des, esp-md5-hmac
tunnel

transform-set CHARLIE
esp-3des, esp-sha-hmac
tunnel

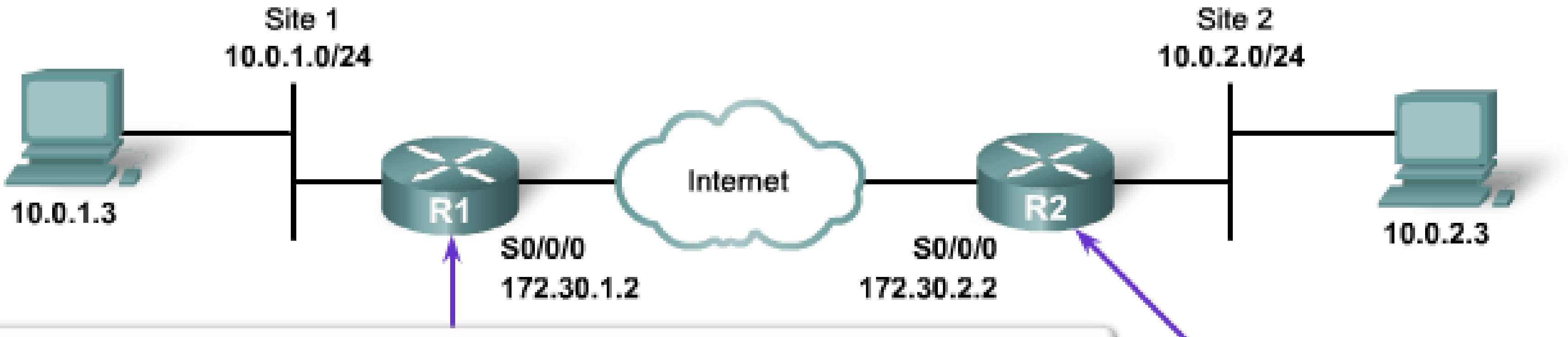
transform-set RED
esp-des
tunnel

transform-set BLUE
esp-des, ah-sha-hmac
tunnel

transform-set YELLOW
esp-3des, esp-sha-hmac
tunnel

- Transform sets are negotiated during IKE Phase 2.
- The 9th attempt found matching transform sets (CHARLIE - YELLOW).

Transform Set Negotiation



```
R1(config)# crypto isakmp key cisco123 address 172.30.2.2
R1(config)# crypto ipsec transform-set MYSET esp-aes 128
R1(cfg-crypto-trans)# exit
R1(config)#
```

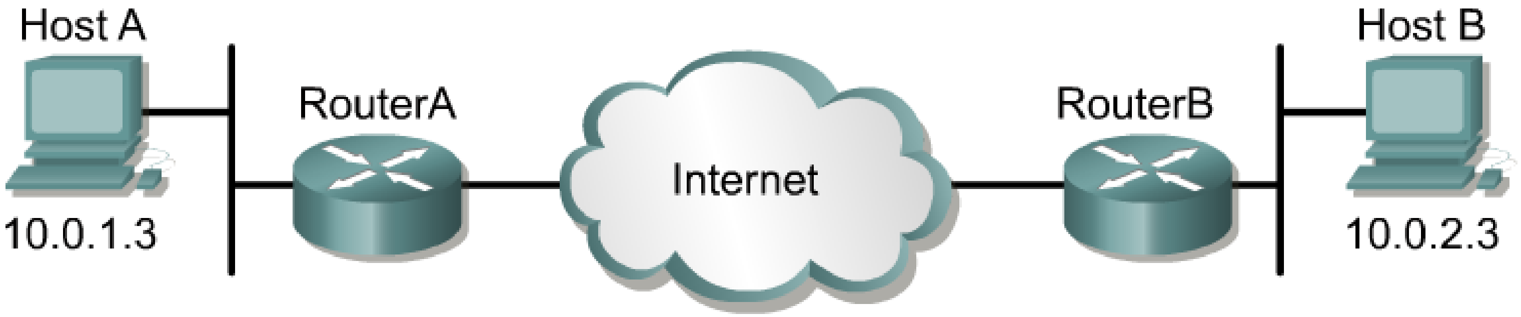
```
R2(config)# crypto isakmp key cisco123 address 172.30.1.2
R2(config)# crypto ipsec transform-set OTHERSET esp-aes 128
R2(cfg-crypto-trans)# exit
```

Note:

- Peers must share the same transform set settings.
- Names are only locally significant.

Configure Security Association Lifetimes

- Configures global IPsec lifetime values used when negotiating IPsec security associations.
- IPsec SA lifetimes are negotiated during IKE phase 2.

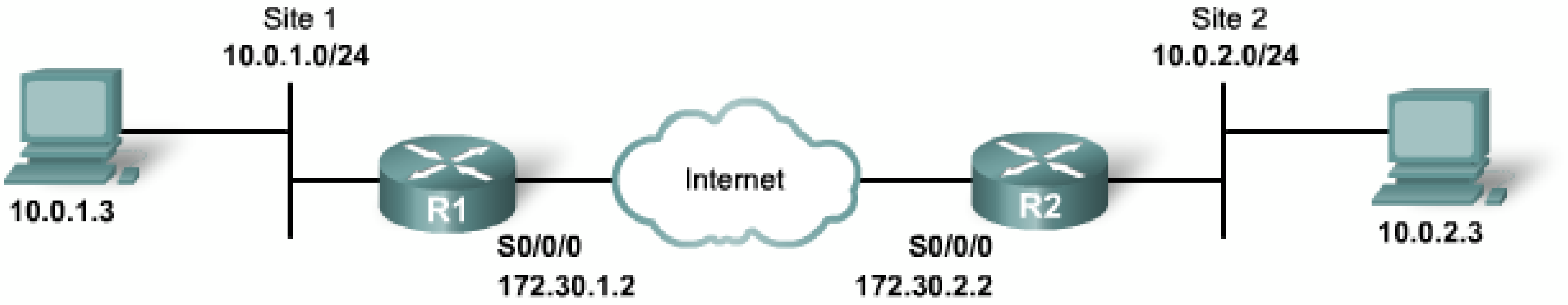


```
router(config)#
```

```
crypto ipsec security-association lifetime
      {seconds seconds | kilobytes kilobytes}
```

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

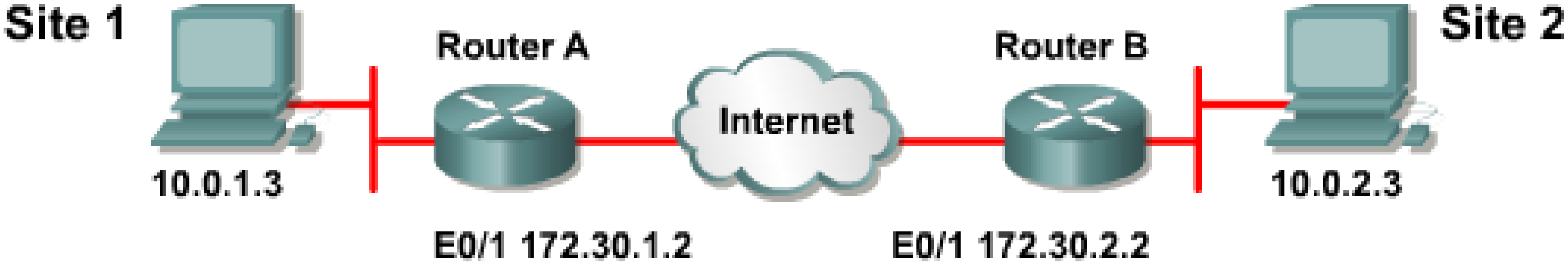
Task 4: Configure Crypto ACLs



```
router(config)#
access-list access-list-number {deny | permit} protocol source source-
wildcard destination destination-wildcard
```

Command	Description
permit	This option causes all IP traffic that matches the specified conditions to be protected by cryptography, using the policy described by the corresponding crypto map entry.
deny	This option instructs the router to route traffic in plaintext.
protocol	This option specifies which traffic to protect by cryptography based on the protocol, such as TCP, UDP, or ICMP. If the protocol is IP, then all IP traffic matching that permit statement is encrypted.
source and *destination*	If the ACL statement is a permit statement, these are the networks, subnets, or hosts between which traffic should be protected. If the ACL statement is a deny statement, then the traffic between the specified source and destination is sent in plaintext.

Configure Symmetrical Peer Crypto ACL



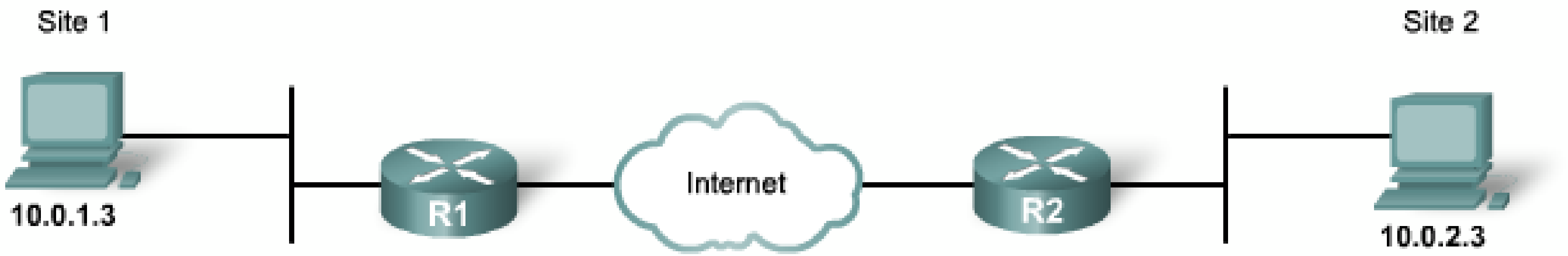
```
RouterA#(config)
```

```
access-list 110 permit tcp 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255
```

```
RouterB#(config)
```

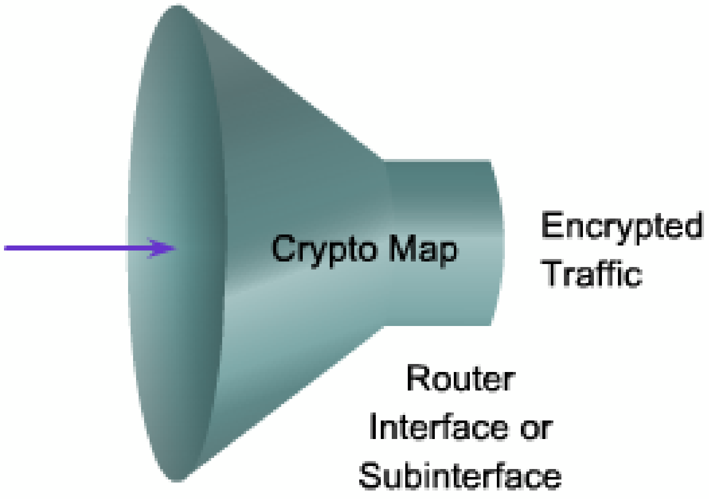
```
access-list 110 permit tcp 10.0.2.0 0.0.0.255 10.0.1.0 0.0.0.255
```

Task 5: Apply the Crypto Map

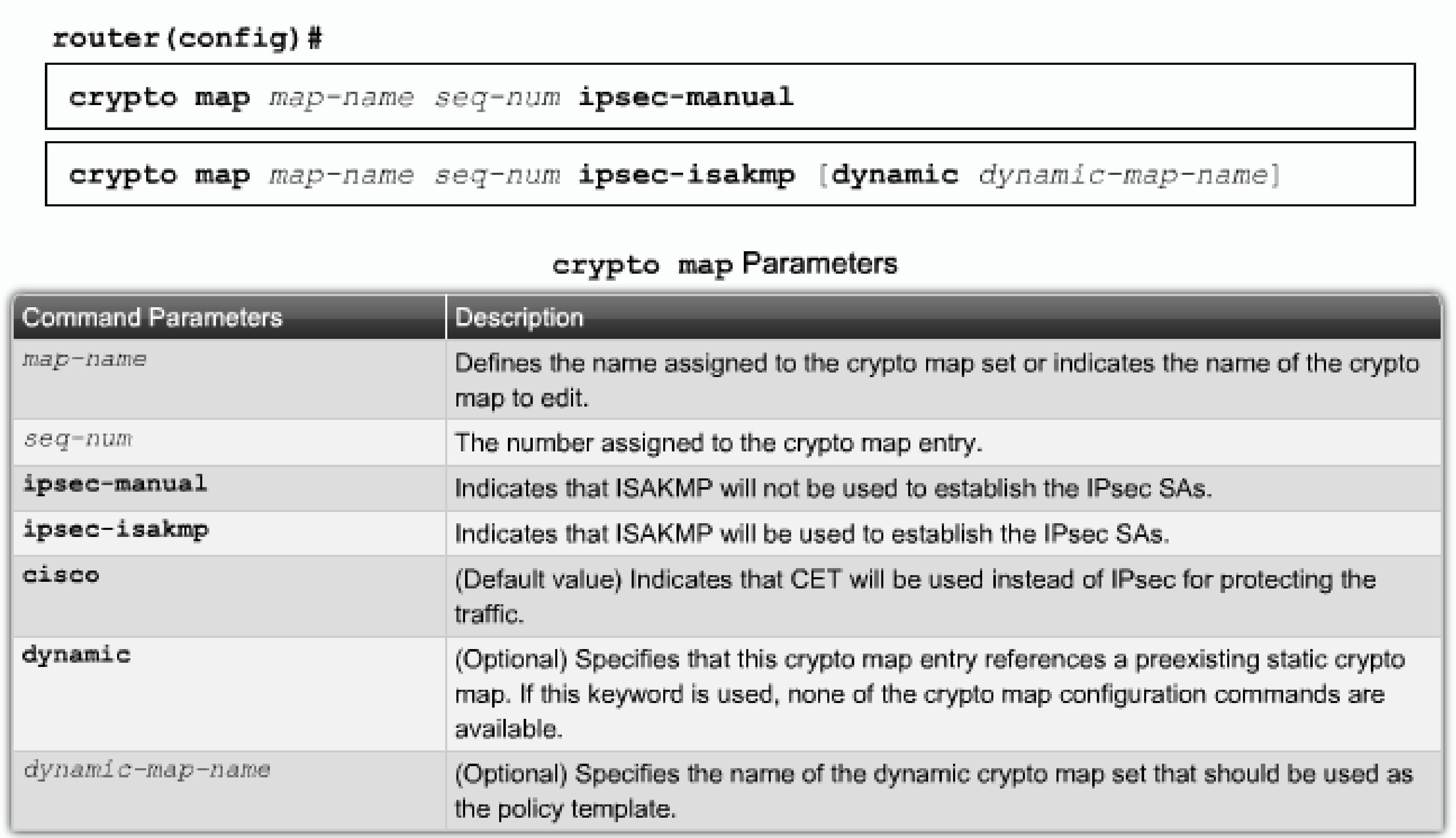


Crypto maps define the following:

- ACL to be used
- Remote VPN peers
- Transform set to be used
- Key management method
- SA lifetimes



Configure IPsec Crypto Maps



```
router(config)#
```

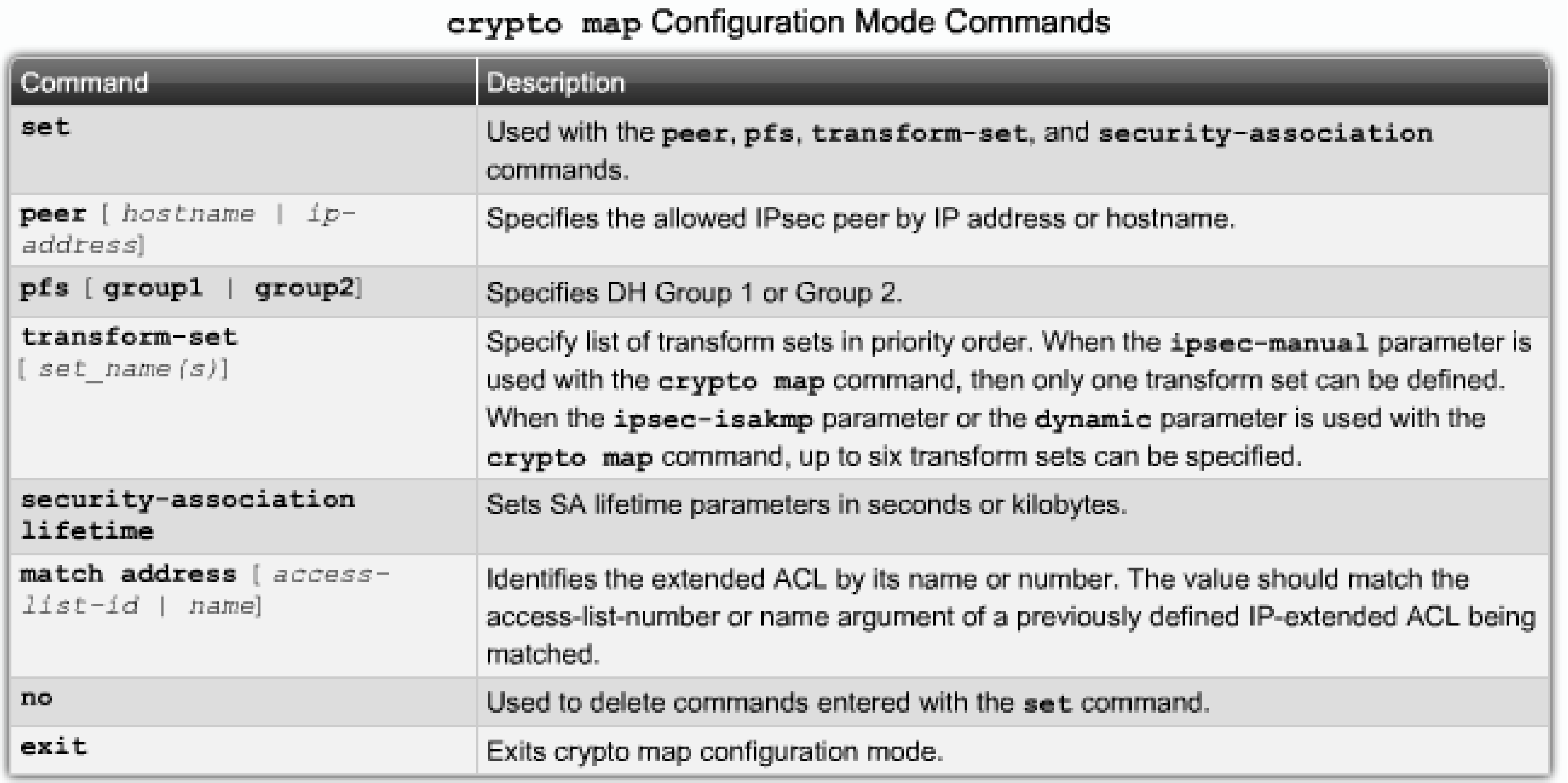
```
crypto map map-name seq-num ipsec-manual
```

```
crypto map map-name seq-num ipsec-isakmp [dynamic dynamic-map-name]
```

`crypto map` Parameters

Command Parameters	Description
map-name	Defines the name assigned to the crypto map set or indicates the name of the crypto map to edit.
seq-num	The number assigned to the crypto map entry.
ipsec-manual	Indicates that ISAKMP will not be used to establish the IPsec SAs.
ipsec-isakmp	Indicates that ISAKMP will be used to establish the IPsec SAs.
cisco	(Default value) Indicates that CET will be used instead of IPsec for protecting the traffic.
dynamic	(Optional) Specifies that this crypto map entry references a preexisting static crypto map. If this keyword is used, none of the crypto map configuration commands are available.
dynamic-map-name	(Optional) Specifies the name of the dynamic crypto map set that should be used as the policy template.

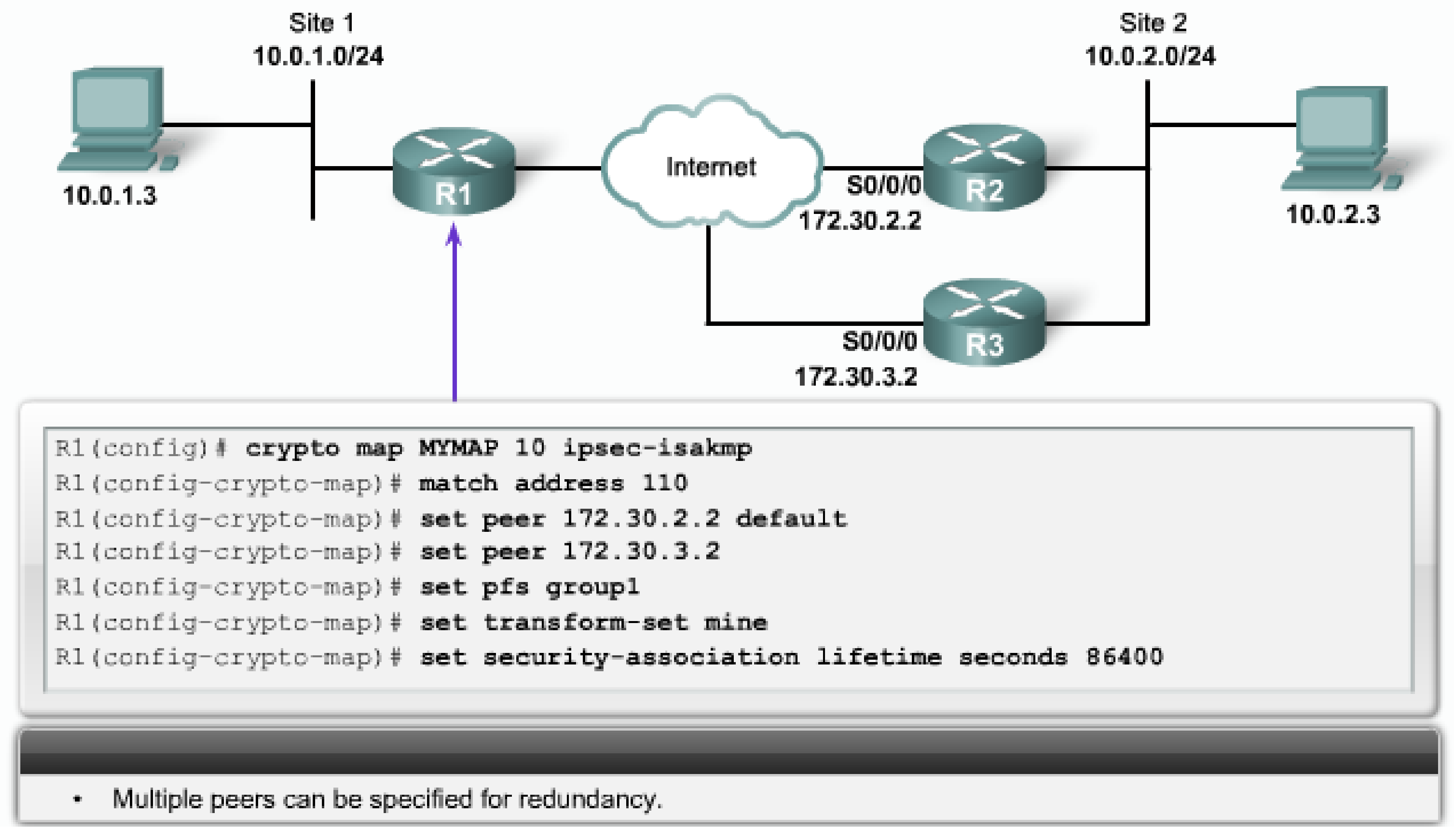
Configure IPsec Crypto Maps



`crypto map` Configuration Mode Commands

Command	Description
`set`	Used with the **`peer`**, **`pfs`**, **`transform-set`**, and **`security-association`** commands.
`peer` [*`hostname`* | *`ip-address`*]	Specifies the allowed IPsec peer by IP address or hostname.
`pfs` [**`group1`** | **`group2`**]	Specifies DH Group 1 or Group 2.
`transform-set` [*`set_name(s)`*]	Specify list of transform sets in priority order. When the **`ipsec-manual`** parameter is used with the **`crypto map`** command, then only one transform set can be defined. When the **`ipsec-isakmp`** parameter or the **`dynamic`** parameter is used with the **`crypto map`** command, up to six transform sets can be specified.
`security-association lifetime`	Sets SA lifetime parameters in seconds or kilobytes.
`match address` [*`access-list-id`* | *`name`*]	Identifies the extended ACL by its name or number. The value should match the access-list-number or name argument of a previously defined IP-extended ACL being matched.
`no`	Used to delete commands entered with the **`set`** command.
`exit`	Exits crypto map configuration mode.

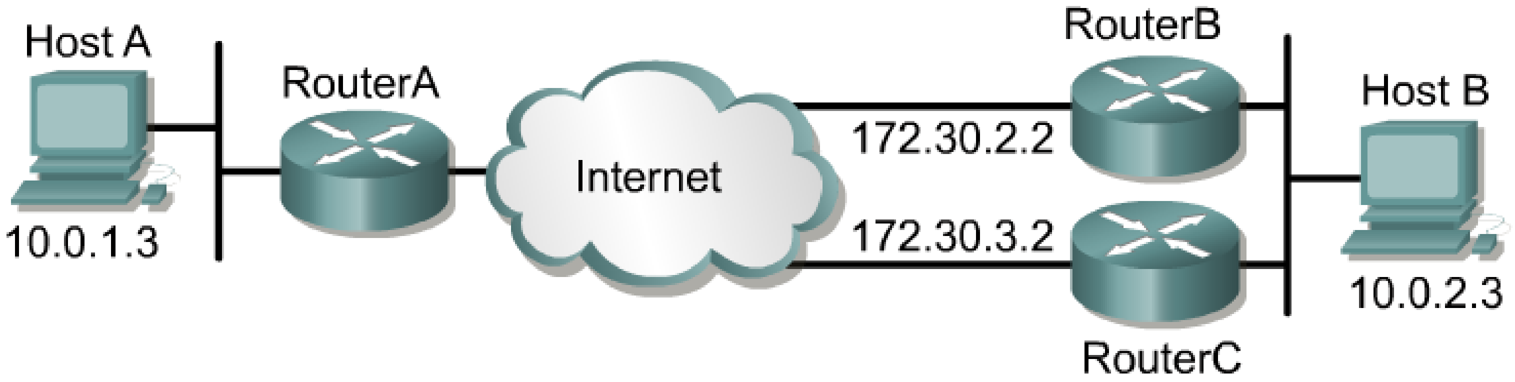
Configure IPsec Crypto Maps



```
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
```

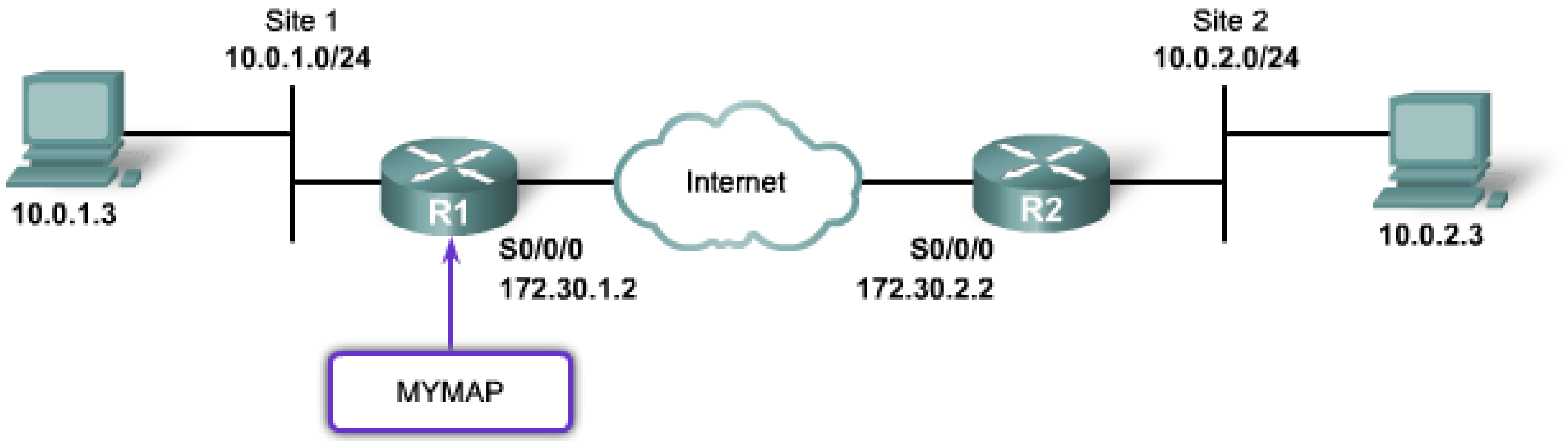
- Multiple peers can be specified for redundancy.

Example Crypto Map Commands



```
RouterA(config)# crypto map MYMAP 110 ipsec-isakmp
RouterA(config-crypto-map)# match address 110
RouterA(config-crypto-map)# set peer 172.30.2.2
RouterA(config-crypto-map)# set peer 172.30.3.2
RouterA(config-crypto-map)# set transform-set MINE
RouterA(config-crypto-map)# set security-association lifetime 86400
```

Applying Crypto Maps to Interfaces



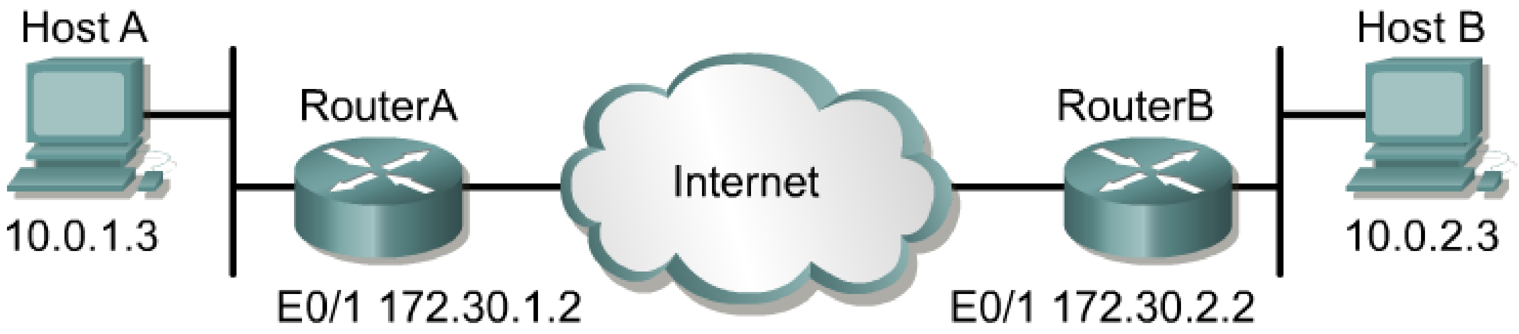
```
router(config-if)#
```

```
crypto map map-name
```

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

- Applies the crypto map to outgoing interface
- Activates the IPsec policy

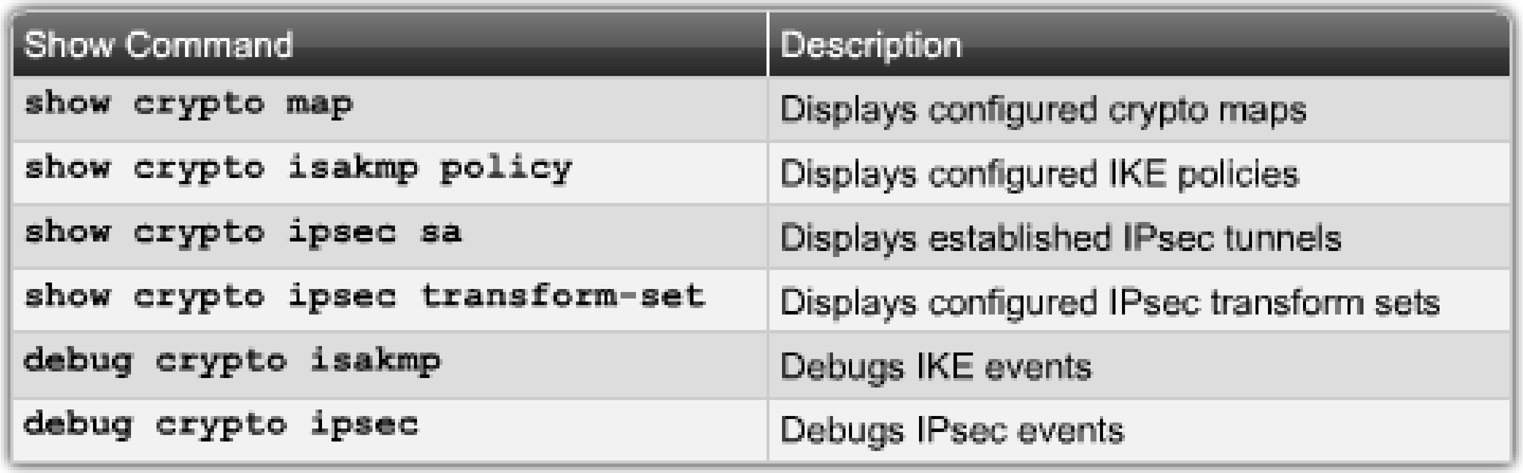
IPsec Configuration Examples



```
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
```

Verify IPsec



Show Command	Description
`show crypto map`	Displays configured crypto maps
`show crypto isakmp policy`	Displays configured IKE policies
`show crypto ipsec sa`	Displays established IPsec tunnels
`show crypto ipsec transform-set`	Displays configured IPsec transform sets
`debug crypto isakmp`	Debugs IKE events
`debug crypto ipsec`	Debugs IPsec events

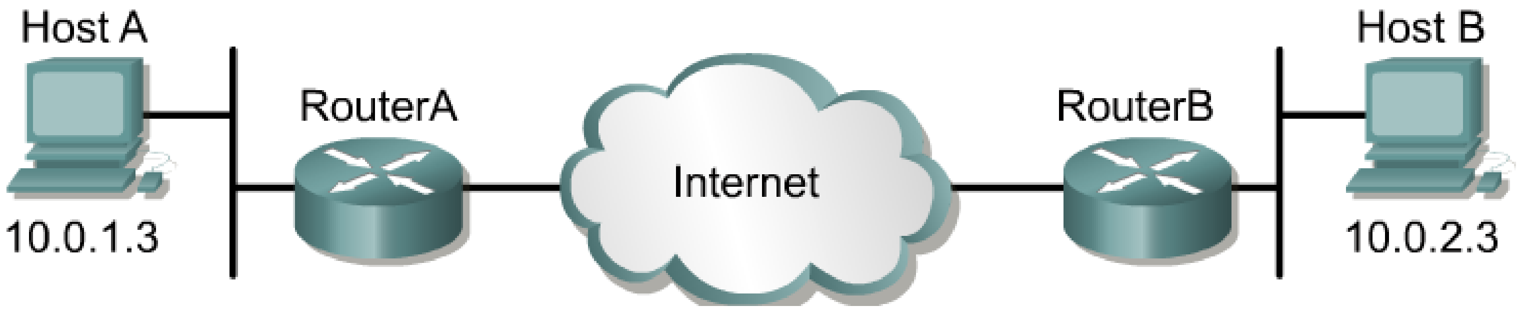
`clear` commands

- Clears IPsec Security Associations in the router database.

```
Router#
```

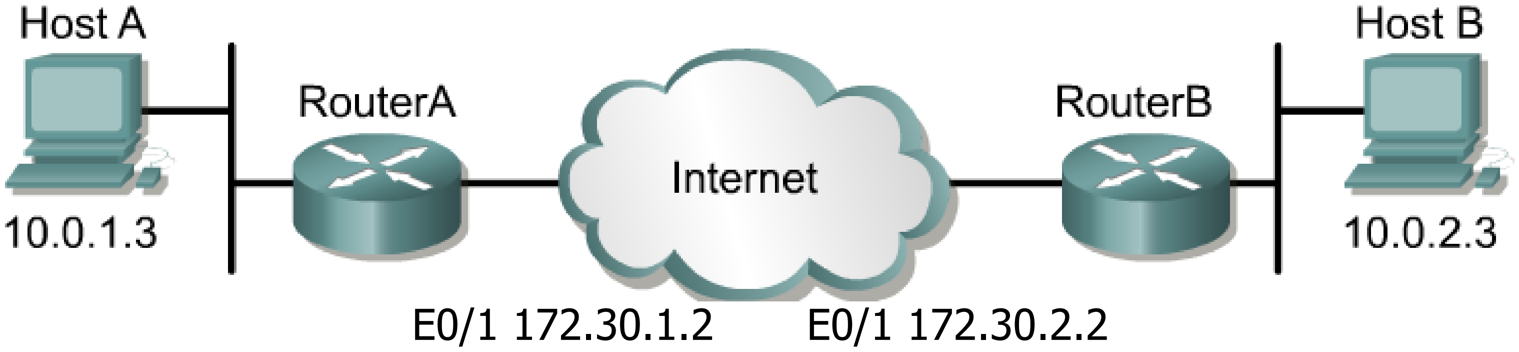
```
clear crypto sa
clear crypto sa peer <IP address | peer name>
clear crypto sa map <map name>
clear crypto sa entry <destination-address protocol spi>
```

View Policy



```
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-share
      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
```

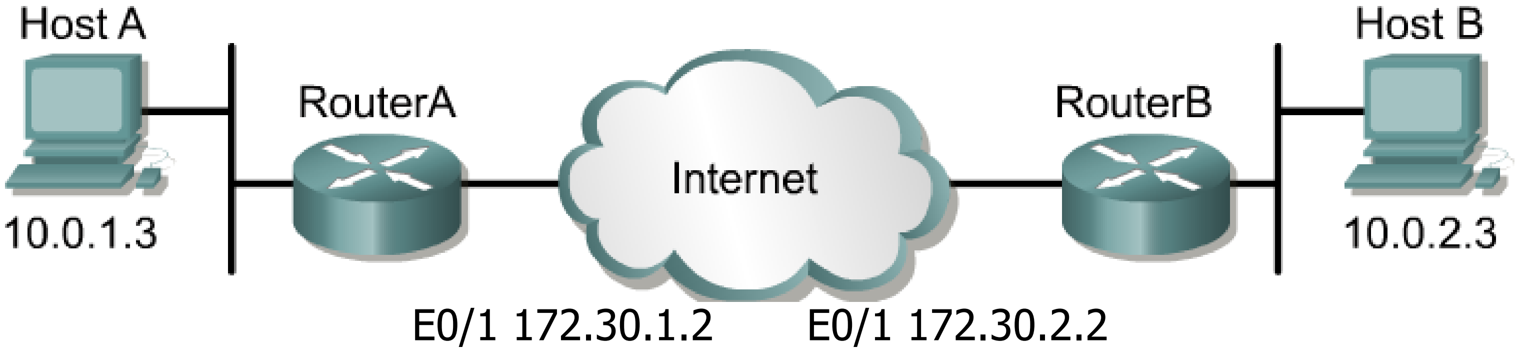
View Defined Sets



```
RouterA# show crypto ipsec transform-set MY-SET
Transform set MY-SET: { esp-des }
will negotiate = { Tunnel, },
```

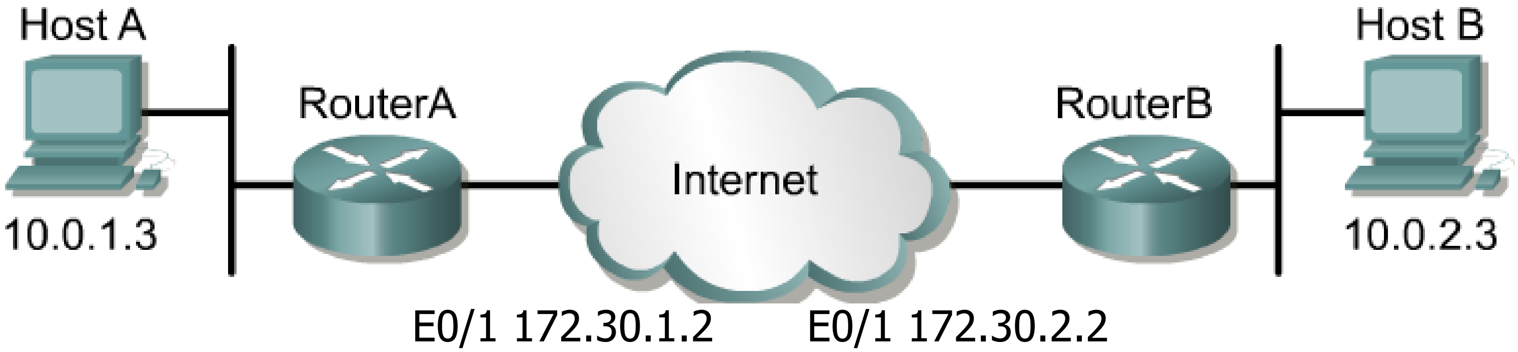
Display Phase 1 SA

- QM_IDLE (quiescent state) indicates that an ISAKMP SA exists but is idle.
- The router will remain authenticated with its peer and may be used for subsequent quick mode (QM) exchanges.



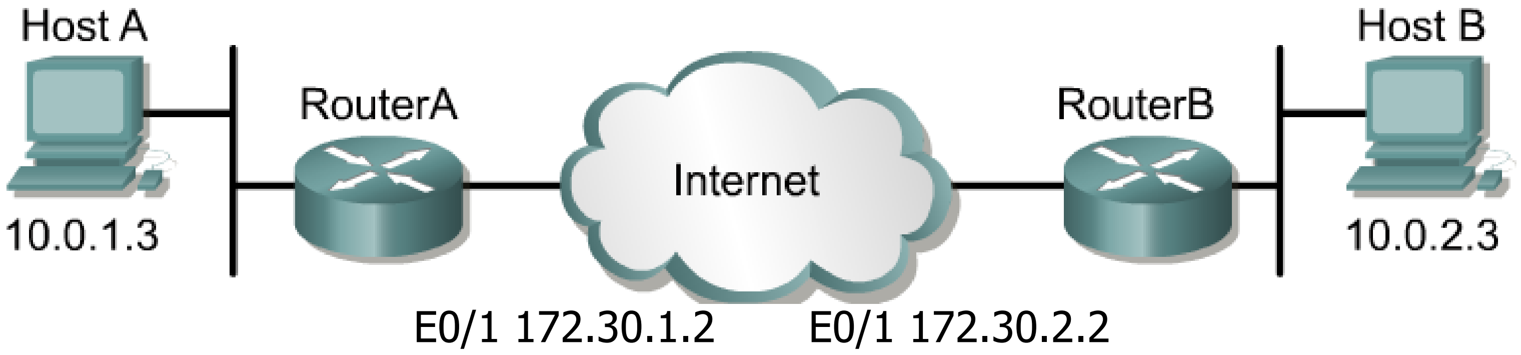
```
RouterA# show crypto isakmp sa

dst          src          state        conn-id           slot
172.30.2.2   172.30.1.2   QM_IDLE      47                5
```

View Crypto IPsec SA



```
RouterA# show crypto ipsec sa
interface: Ethernet0/1
     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
```

View Configured Crypto Maps



```
RouterA# show crypto map
Crypto Map “MYMAP" 10 ipsec-isakmp
          Peer = 172.30.2.2
          Extended IP access list 102
            access-list 102 permit ip host 172.30.1.2 host 172.30.2.2
          Current peer: 172.30.2.2
          Security association lifetime: 4608000 kilobytes/3600 seconds
          PFS (Y/N): N
          Transform sets={ MINE, }
```

Crypto System Error Messages for ISAKMP

- To display debug messages about all IPsec actions, use the global command `debug crypto ipsec`.
- To display debug messages about all ISAKMP actions, use the global command `debug crypto isakmp`.

Crypto System Error Messages for ISAKMP

- ISAKMP SA with the remote peer was not authenticated.

```
%CRYPTO-6-IKMP_SA_NOT_AUTH: Cannot accept Quick Mode exchange
from %15i if SA is not authenticated!
```

- ISAKMP peers failed protection suite negotiation for ISAKMP.

```
%CRYPTO-6-IKMP_SA_NOT_OFFERED: Remote peer %15i responded with
attribute [chars] not offered or changed
```

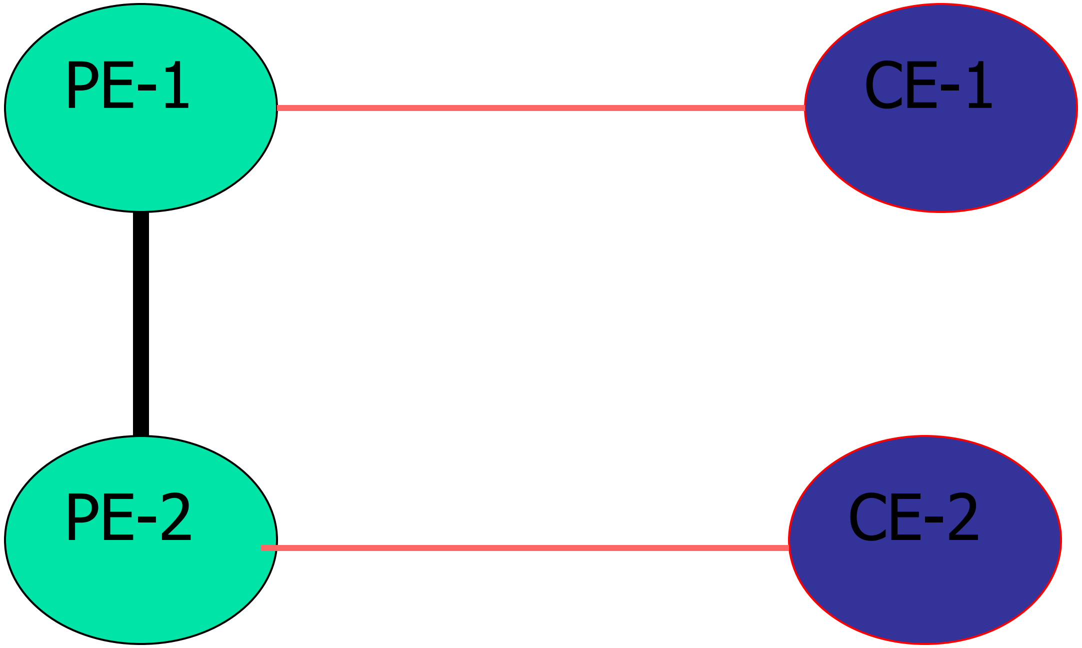
Crypto System Error Messages for ISAKMP

- This is an example of the Main Mode error message.
- The failure of Main Mode suggests that the Phase I policy does not match on both sides.

```
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 150.150.150.1
```

- Verify that the Phase I policy is on both peers and ensure that all the attributes match.
  - Encryption: DES or 3DES
  - Hash: MD5 or SHA
  - Diffie-Hellman: Group 1 or 2
  - Authentication: rsa-sig, rsa-encr or pre-share

Site-to-Site IPSec VPN



Site-to-Site IPSec VPN

Check from CE

- Check the route 200.0.X.1/32 among CEs. It should be exist on the Routing Table.
- Check connectivity between CEs with ip address 200.0.x.1

Configuring IPsec

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.

Configure the IKE Phase 1 Policy(ISAKMP Policy)

```
CE-1(config)# crypto isakmp policy <policy_number>
CE-1(config-isakmp)#encryption 3des
CE-1(config-isakmp)#authentication pre-share
CE-1(config-isakmp)#group 2
CE-1(config-isakmp)#hash md5
CE-1(config-isakmp)#lifetime 86400
```

Configure the IKE Phase 1 Policy(ISAKMP Policy)

```
CE-2(config)# crypto isakmp policy <policy_number>
CE-2(config-isakmp)#encryption 3des
CE-2(config-isakmp)#authentication pre-share
CE-2(config-isakmp)#group 2
CE-2(config-isakmp)#hash md5
CE-2(config-isakmp)#lifetime 86400
```

Set the ISAKMP Identity and Key

- The ISAKMP identity specifies how the IKE Phase 1 peer is identified. Which can be either by ip address or host name.
- The Command to use is:

```
Crypto isakmp identity {address | hostname | dn}
```

- By default, a peer's ISAKMP identity is the peer's ip address. If you decide to change the default just keep in mind that it is best to always be consistent across your entire IPSec-protected network in the way you choose to define a peer's identity

Set the ISAKMP pre-shared key

The ISAKMP pre-shared key is the shared secret that is used to authenticate the peers in the Diffie-Hellman exchange.The command to use is:

```
Crypto isakmp key <pre-shared-key> address <peer ip address>
```

Configure the IPSec AH and ESP Parameters

Create a transform set with name Cisco2Cisco in transport mode and using esp-3des and esp-md5-hmac

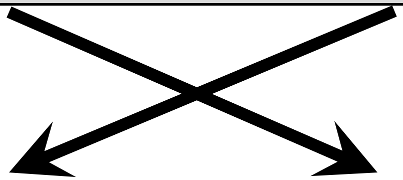
The AH and ESP parameters are configured with the following commands:

crypto ipsec transform-set <transform-set-name> <transform 1> <transform 2> mode [tunnel | transport] crypto ipsec security-association lifetime seconds seconds

Eg. Crypto ipsec transform-set cisco2cisco

Create a Crypto ACL

```
CE-1#(config)
access-list 110 permit tcp 200.0.X.1 0.0.0.1 200.0.X1.1 0.0.0.0
```



```
CE-2#(config)
access-list 110 permit tcp 200.0.X1.0 0.0.0.0 200.0.X.0 0.0.0.0
```

Create and apply Crypto map

```
CE-1(config)# crypto map MYMAP 110 ipsec-isakmp
CE-1(config-crypto-map)# match address <ACL-Number>
CE-1(config-crypto-map)# set peer CE2
CE-1(config-crypto-map)# set transform-set <Transform-set name>
CE-1(config-crypto-map)# set security-association lifetime 86400
```

```
CE-2(config)# crypto map MYMAP 110 ipsec-isakmp
CE-2(config-crypto-map)# match address <ACL-Number>
CE-2(config-crypto-map)# set peer CE1
CE-2(config-crypto-map)# set transform-set <Transform-set name>
CE-2(config-crypto-map)# set security-association lifetime 86400
```

Applying Crypto Maps to Interfaces

```
CE-1(config)# interface eth x / x
CE-1(config-if)# description Connected to PE-1 interface X/X
CE-1(config-if)# crypto map MYMAP
```

```
CE-2(config)# interface eth x / x
CE-2(config-if)# description Connected to PE-2 interface X/X
CE-2(config-if)# crypto map MYMAP
```