Réseaux Privés Virtuels
Virtual Private Network

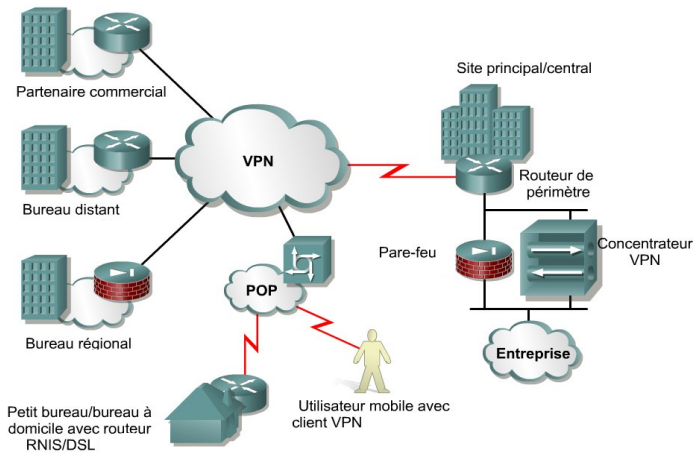
Philippe.Arnould@univ-pau.fr

Plan

- Définitions
- IPSEC
- MPLS
- Exemples
- Bibliographies

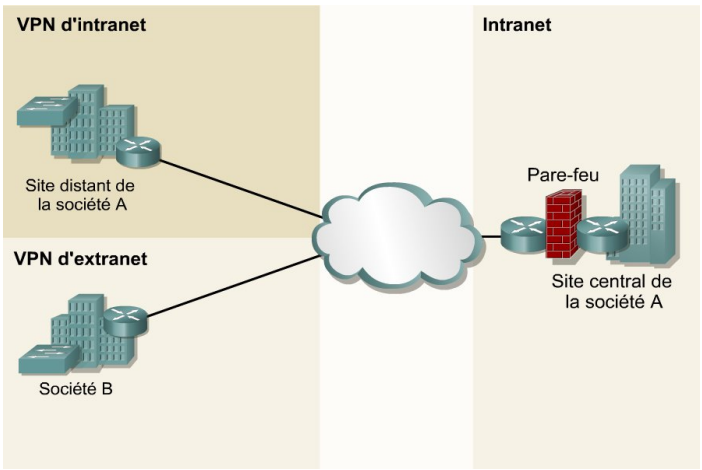
2

Définitions



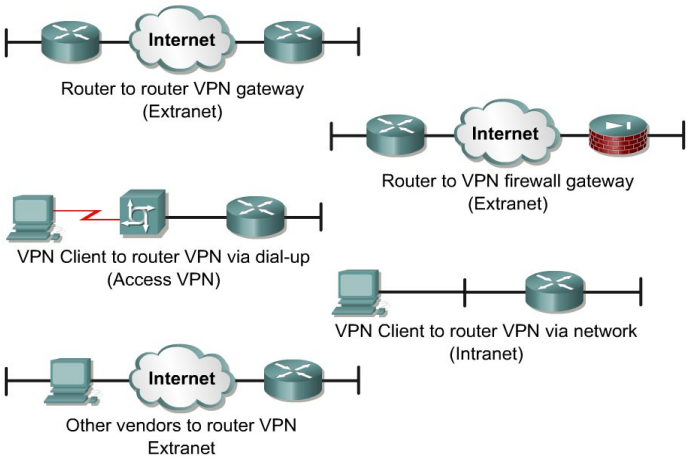
3

Définitions



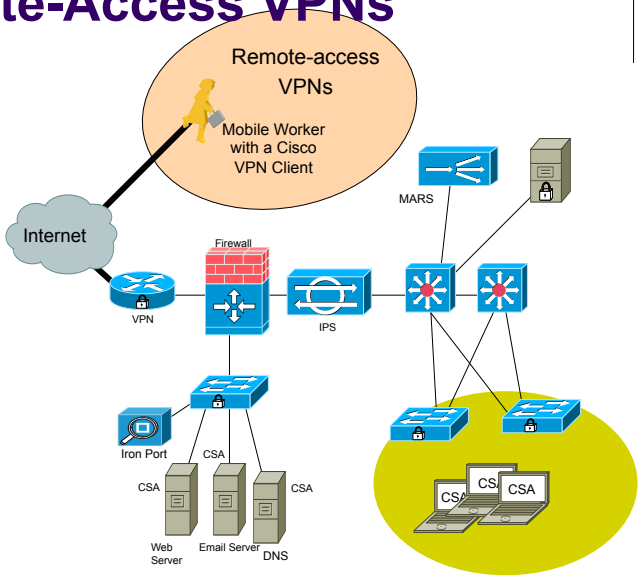
4

Définitions

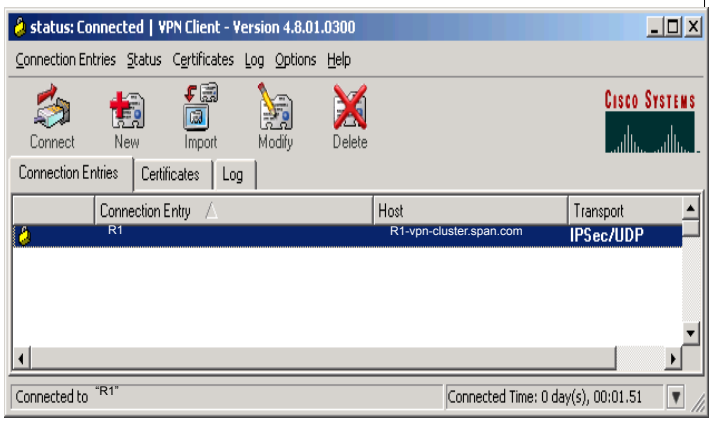


5

Remote-Access VPNs



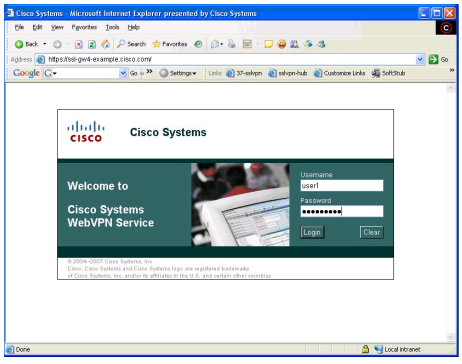
VPN Client Software



In a remote-access VPN, each host typically has Cisco VPN Client software

Cisco IOS SSL VPN

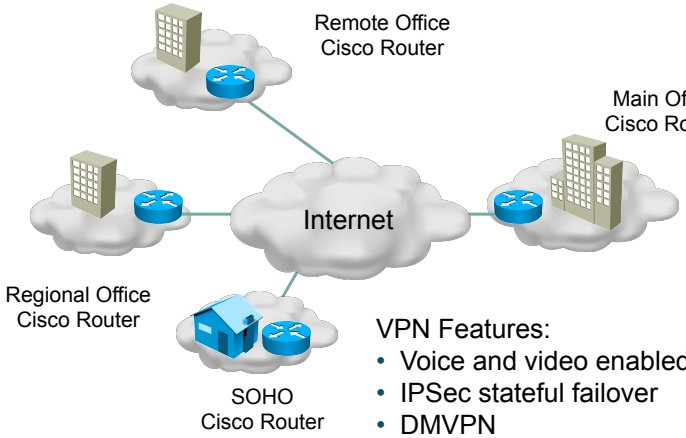
- Provides remote-access connectivity from any Internet-enabled host
- Uses a web browser and SSL encryption
- Delivers two modes of access:
  - Clientless
  - Thin client



Cisco VPN Product Family

Product Choice	Remote-Access VPN	Site-to-Site VPN
Cisco VPN-Enabled Router	Secondary role	Primary role
Cisco PIX 500 Series Security Appliances	Secondary role	Primary role
Cisco ASA 5500 Series Adaptive Security Appliances	Primary role	Secondary role
Cisco VPN 3000 Series Concentrators	Primary role	Secondary role
Home Routers	Primary role	

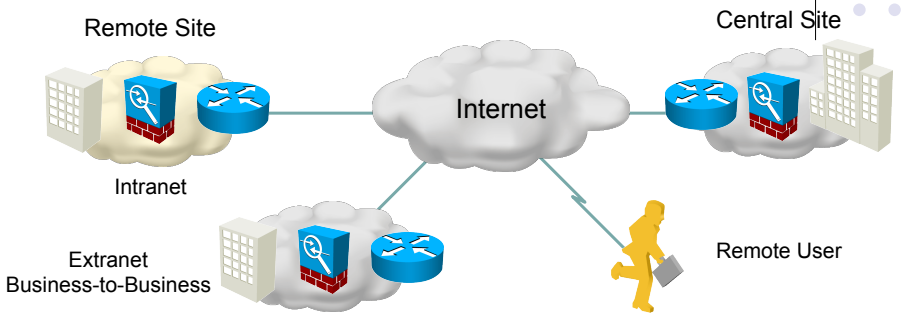
Cisco VPN-Optimized Routers



VPN Features:

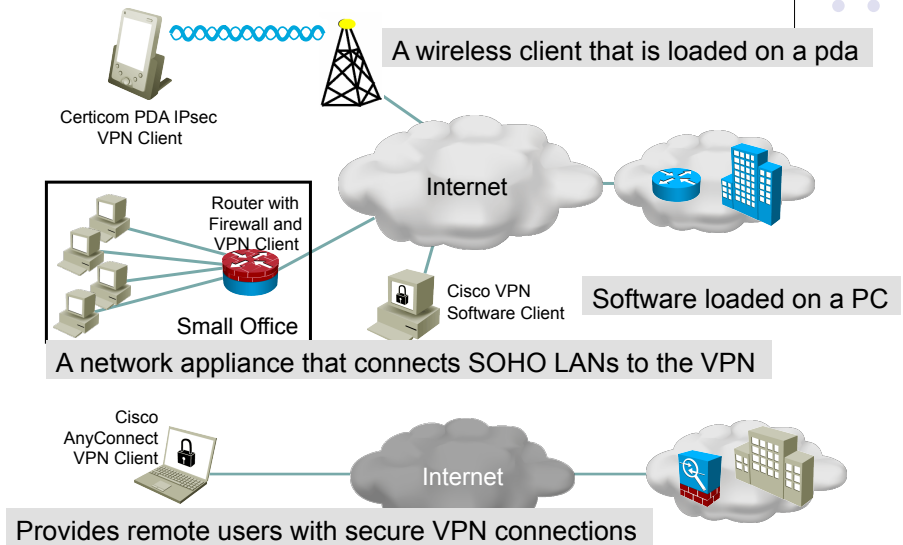
- Voice and video enabled VPN (V3PN)
- IPSec stateful failover
- DMVPN
- IPSec and Multiprotocol Label Switching (MPLS) integration
- Cisco Easy VPN

Cisco ASA 5500 Series Adaptive Security Appliances



- Flexible platform
- Resilient clustering
- Cisco Easy VPN
- Automatic Cisco VPN
- Cisco IOS SSL VPN
- VPN infrastructure for contemporary applications
- Integrated web-based management

IPSec Clients



Hardware Acceleration Modules

- AIM
- Cisco IPSec VPN Shared Port Adapter (SPA)
- Cisco PIX VPN Accelerator Card+ (VAC +)
- Enhanced Scalable Encryption Processing (SEP-E)

Cisco IPsec VPN SPA

Définitions

- Deux catégories de VPN :
  - “CE Based” : VPN construits à l’aide des passerelles installées sur les sites à interconnecter
  - “Network Based” construit au niveau du transport

14

VPN “CE Based”

- Implantées sur des passerelles qui se mettent en coupure entre les sites à interconnecter et le réseau de transport public comme internet
- Passerelles : routeur ou firewall gérées par l’opérateur ou l’entreprise
- Rôle : encapsuler le trafic sortant dans des datagrammes IP -> tunnels virtuels
- Types de tunnels:
  - Tunnels Ipsec qui garantissent l’intégrité, l’authentification et/ou la confidentialité des échanges
  - Tunnels dont la composante de sécurité n’est pas essentielle

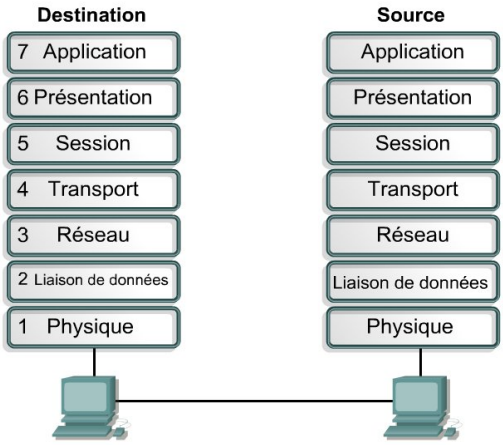
15

VPN “Network Based”

- Les fonctions sont implémentées sur les équipements au cœur du réseau.
- Permet le partage des coûts d’infrastructure réseau et de nouveaux services (QoS)
- MPLS (Multi-Protocole Label Switching)
- Frame Relay ou ATM

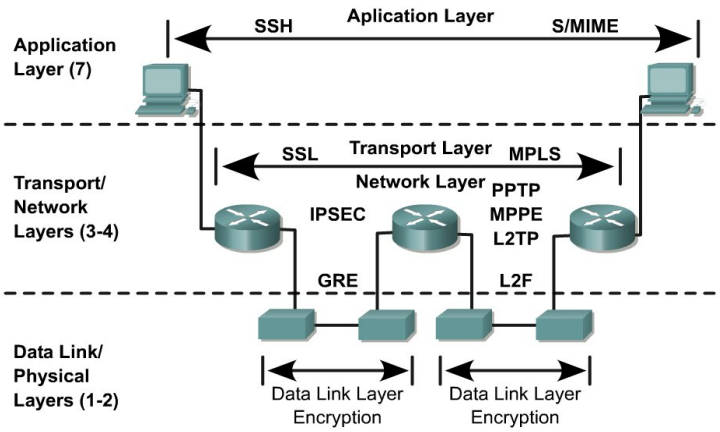
16

Protocoles



17

Protocoles VPN



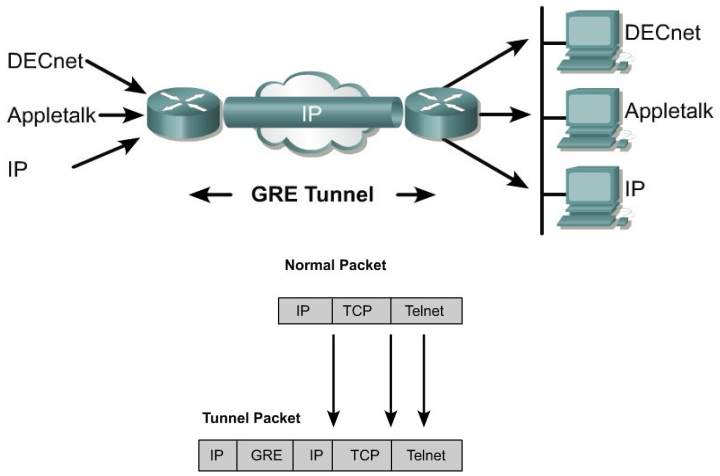
18

Protocoles

	Description	Standard
GRE	Generic Routing Encapsulation	RFC1701 and 2784
IPSec	Internet Protocol Security	RFC2401
L2F	Layer 2 Forwarding	Cisco
L2TP	Layer 2 Tunneling Protocol	RFC 2661
MPLS	Multiprotocol Label Switching	RFC 2547
PPTP	Point-To-Point Tunneling Protocol	Microsoft

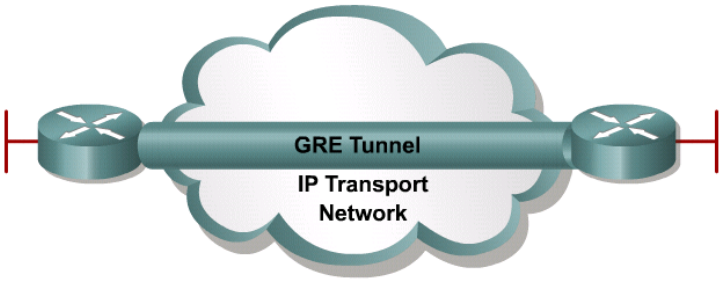
19

Protocole GRE



20

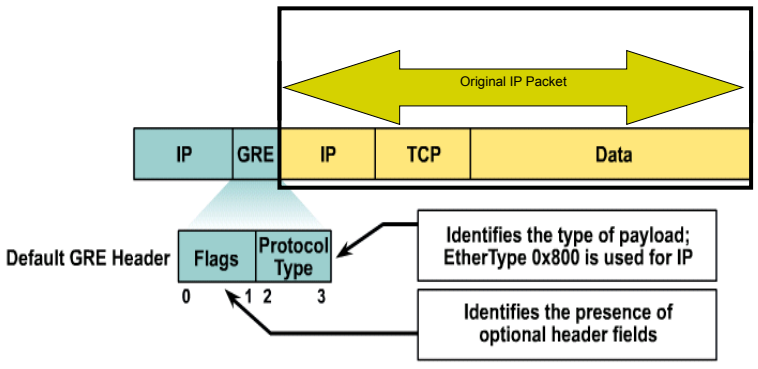
GRE VPN Overview



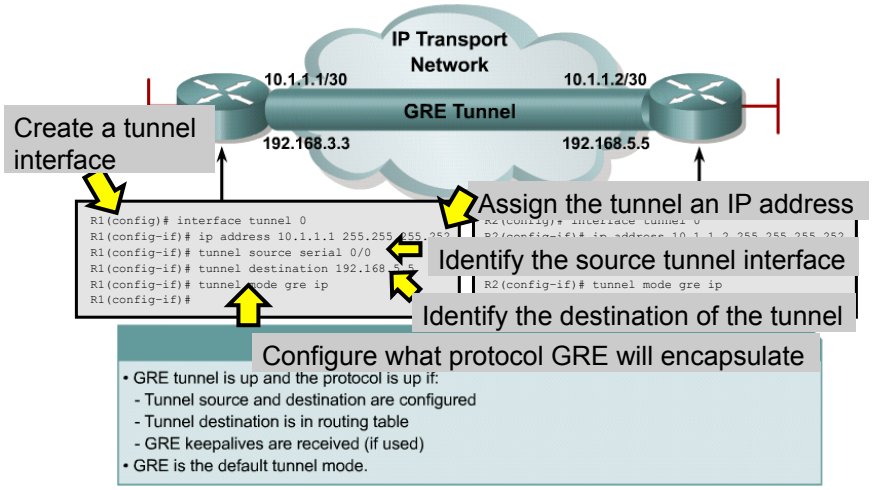
OSI Layer 3 tunneling protocol:
- Encapsulates a wide variety of protocol packet types inside IP tunnels
- Creates a virtual point-to-point link to Cisco routers at remote points over an IP internetwork
- Uses IP for transport
- Uses an additional header to support any other OSI Layer 3 protocol as payload (for example, IP, IPX, AppleTalk)

Encapsulation

Encapsulated with GRE

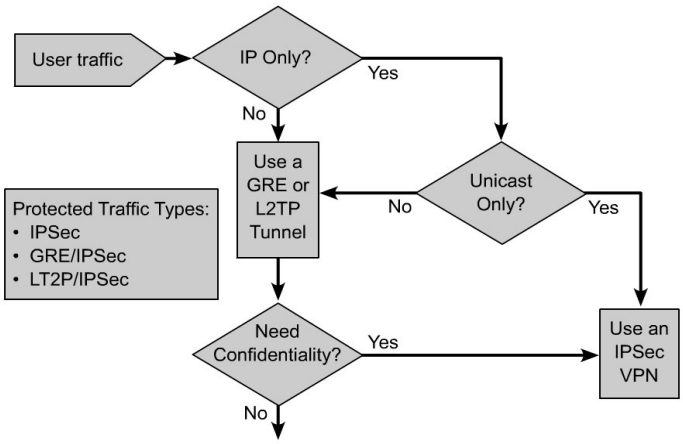


Configuring a GRE Tunnel



- GRE tunnel is up and the protocol is up if:
  - Tunnel source and destination are configured
  - Tunnel destination is in routing table
  - GRE keepalives are received (if used)
- GRE is the default tunnel mode.

Choix du protocole



24

IPSEC

- IP Security est un ensemble de protocoles spécifiant deux aspects :
  - L'encapsulation des datagrammes IP dans d'autres datagrammes IP de manière à fournir des services de sécurité classique :
    - Intégrité
    - Confidentialité
    - Authentification
  - La négociation des clés et des associations de sécurité utilisées lors de l'encapsulation.

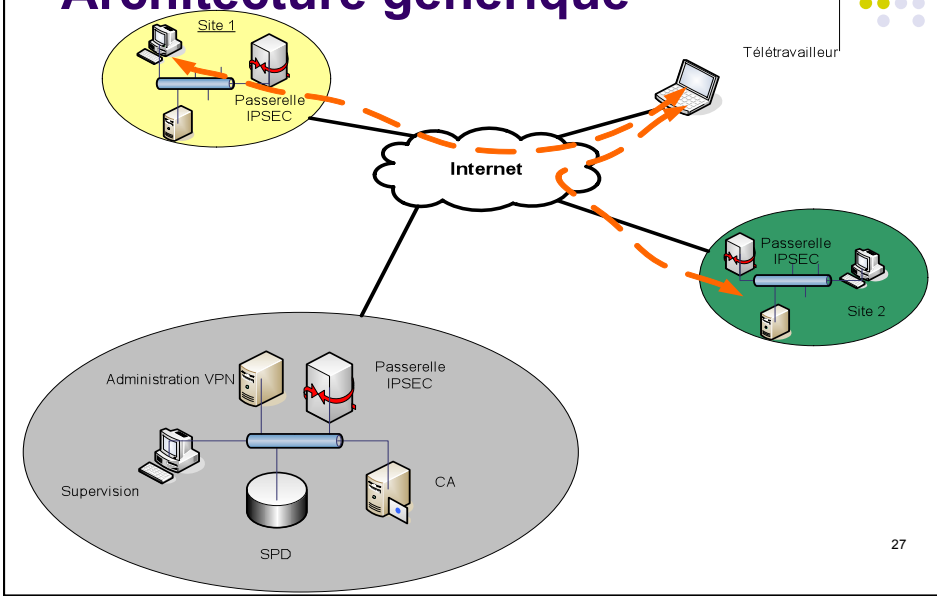
25

IPSEC
Architecture générique

- Passerelles de sécurité
- Une base de données des règles de sécurité (SPD security policy database) qui spécifie pour chaque datagramme IP s'il est nécessaire de le protéger et si oui les services de sécurité de mise en œuvre.
- Un équipement d'administration du service VPN
- Un serveur de certificats (CA certfication Authority)
- Un serveur de supervision (état, alarmes,..)
- Un équipement de journalisation (log)

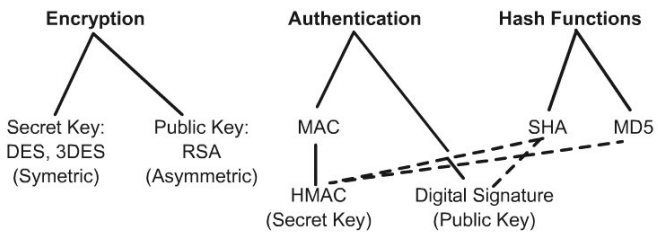
26

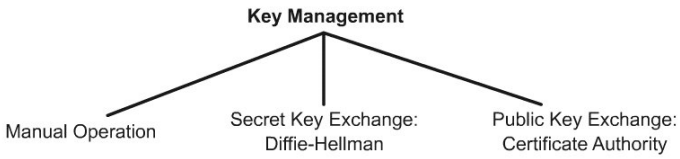
IPSEC
Architecture générique



27

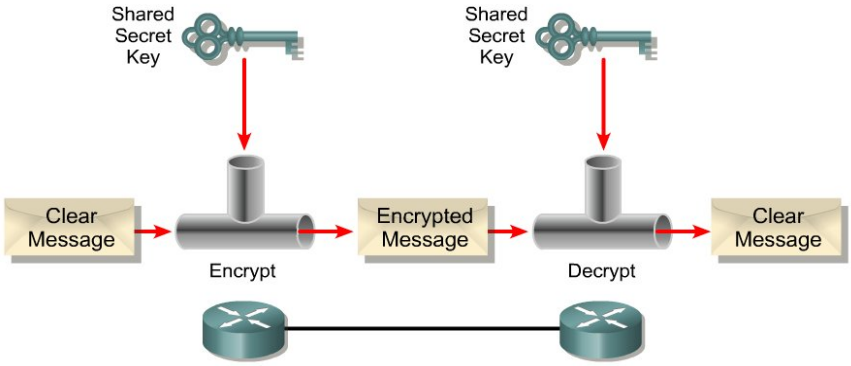
Cryptographie





28

Chiffrement symétrique



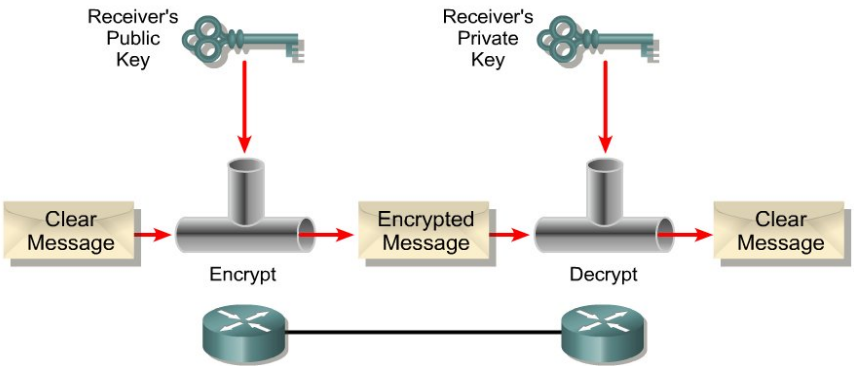
29

Chiffrement symétrique

- DES
  - IBM 1977
  - http://cryptage.online.fr/html/DES/des.html
- 3DES
  - algorithme DES appliqué trois fois
- **A**dvanced **E**ncryption **S**tandard

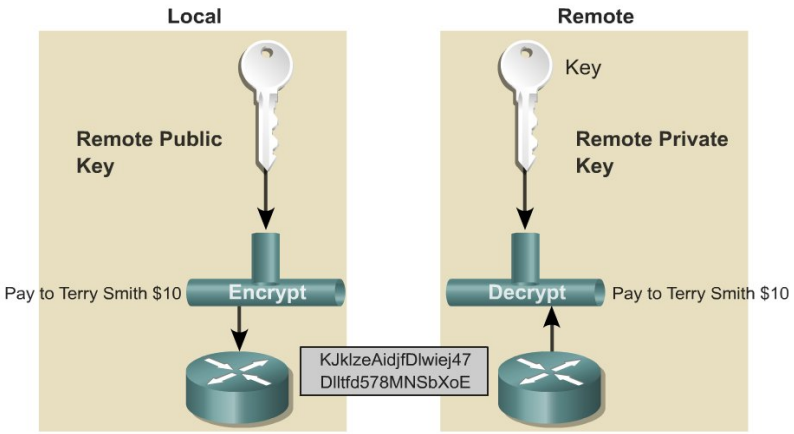
30

Chiffrement asymétrique



31

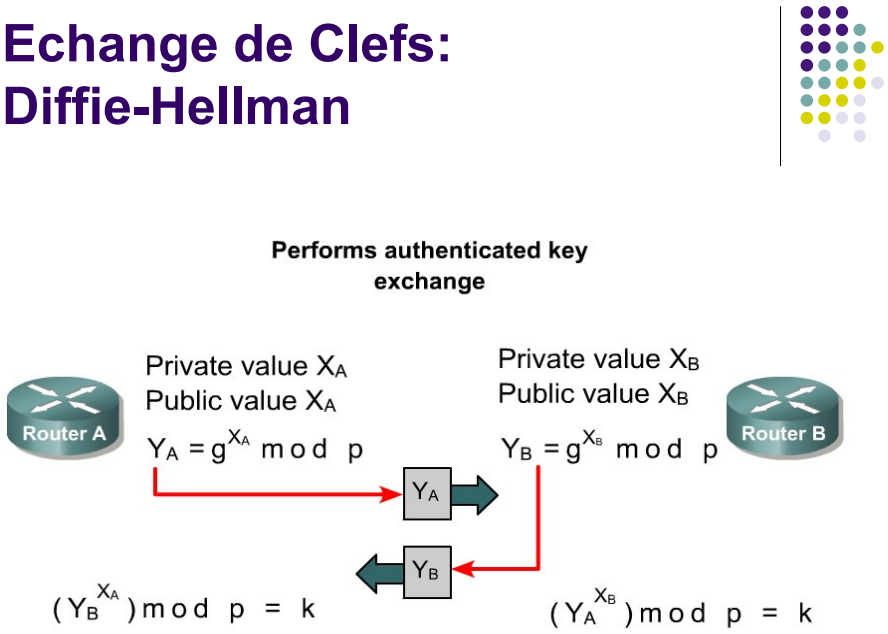
Chiffrement asymétrique Rivest Shamir Adleman



http://cryptage.online.fr/html/RSA/RSA.html

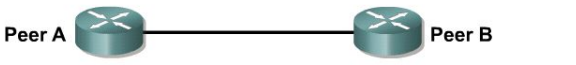
32

Echange de Clefs: Diffie-Hellman



33

Echange de Clefs: Diffie-Hellman



Peer A

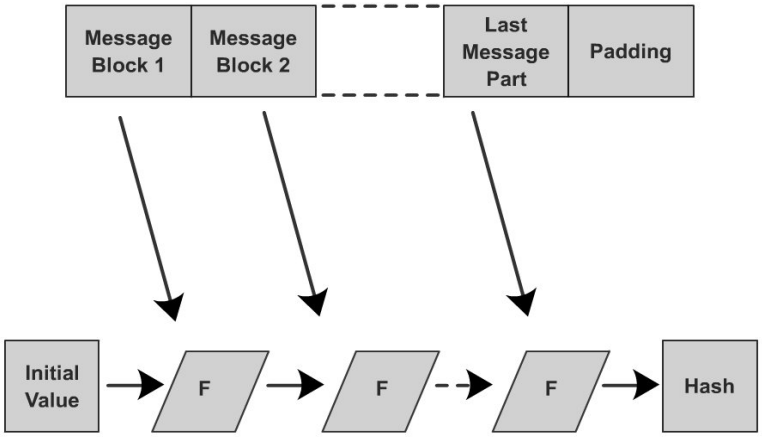
1. Generate large integer p. Send p to Peer B. Receive q. Generate g.
2. Generate private key X_A
3. Generate public key $Y_A = g \wedge X_A \bmod p$
4. Send public key Y_A
5. Generate shared secret number $ZZ = Y_B \wedge X_A \bmod p$
6. Generate shared secret key from ZZ (56-bit for DES, 168-bit for 3DES)

Peer B

1. Generate large integer q. Send q to Peer A. Receive p. Generate g.
2. Generate private key X_B
3. Generate public key $Y_A = g \wedge X_B \bmod p$
4. Send public key Y_B
5. Generate shared secret number $ZZ = Y_A \wedge X_B \bmod p$
6. Generate shared secret key from ZZ (56-bit for DES, 168-bit for 3DES)

34

Intégrité des données : «Hashage»



35

Intégrité des données : “Hashage”

Pay to Terry Smith $10
4ehIDx67Nmop9

Pay to Alex Jones $100
12hqPx67NMoX

Match = No Changes
No Match = Alterations

36

Intégrité des données : “Hashage”

- MD5
  - Message Digest 128 bits
  - inventé par le Ron Rivest (R de RSA)
  - RFC 1321 http://www.faqs.org/rfcs/rfc1321.html
- SHA
  - **S**ecure **H**ash **A**lgorithme
  - 160 bits plus sûr que le MD5
  - http://www.itl.nist.gov/fipspubs/fip180-1.htm

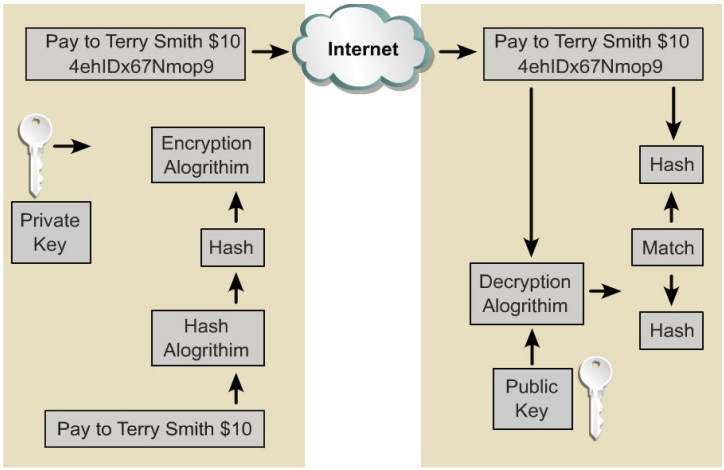
37

Signatures numériques avec clef publique

- garantir qu'un message a bien été émis par celui qui prétend l'avoir émis et que le message reçu est identique à celui émis
- utilisation d'un algorithme asymétrique dont la clef privée - propriété exclusive du signataire - sert à chiffrer une empreinte faite à partir du message original à transmettre.
- Le message et sa signature sont ensuite transmis au destinataire.
- Le destinataire connaît la clef publique du signataire (largement diffusée) qui seule correspond à la clef privée utilisée pour le chiffrement.
- Cette clef lui sert à vérifier la signature en déchiffrant l'empreinte du message, en recalculant une autre empreinte à partir du message reçu et en comparant les 2 résultats qui se doivent d'être identiques.
- Cette méthode permet :
  - d'identifier la provenance du document (la clef privée ultra protégée en est la seule garantie)
  - d'en assurer l'intégrité par la méthode de comparaison des empreintes (voir les techniques de hashage)
- RSA commercial et DSA pour le gouvernement américain

38

Signatures numériques avec clé publique



39

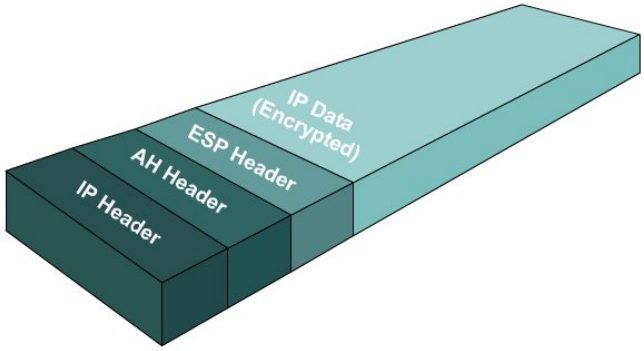
IPSEC

RFC 2401
http://www.faqs.org/rfcs/rfc2401.html

IPSec Framework	Choice 1	Choice 2
IPSec protocol	ESP	ESP +AH
Encryption	DES	3DES
Authentication	MD5	SHA
Diffie -Hellman	DH1	DH2

40

IPSEC



41

IPSEC

Authentication Header

- Ensures data integrity
- Provides origin authentication - ensures packets definitely came from peer router
- Uses keyed-hash mechanism
- Does NOT provide confidentiality (no encryption)
- Provides optional replay protection

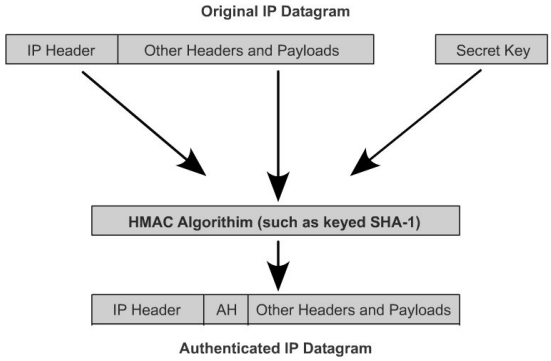
Encapsulating Security Payload

- Data confidentiality (encryption)
- Limited traffic flow confidentiality
- Data integrity
- Optional data origin authentication
- Anti-replay protection
- Does not protect IP header

42

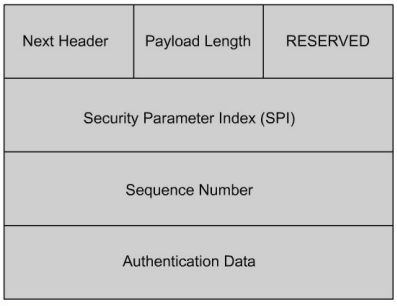
Authentication Header

RFC 2402



43

Authentication Header

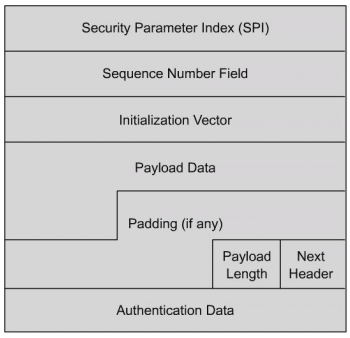


- A 32-bit Security Parameter Index (SPI) value shows the Security Association (SA) used for this packet
- A 64-bit sequence number prevents packet replay
- Authentication data is a HMAC value of the packet

44

Encapsulating security payload

RFC 2406

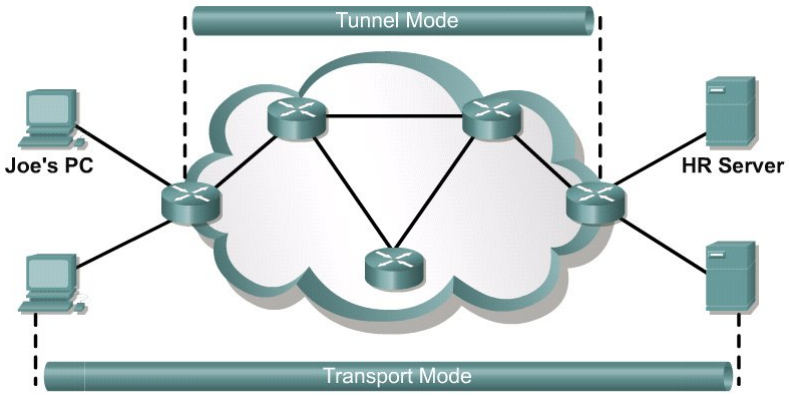


Encryption is done with DES or 3DES. Optional authentication and integrity are provided with HMAC, keyed SHA-1/RFC 2404, or keyed MD5/RFC 2403. There are two different key types contained in the SA:

- Encryption session keys
- HMAC session keys

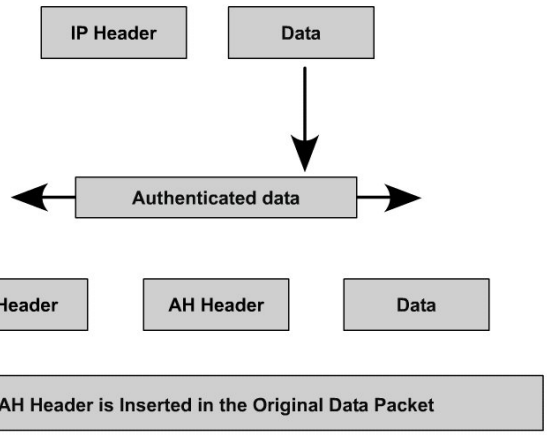
45

Modes IPSEC



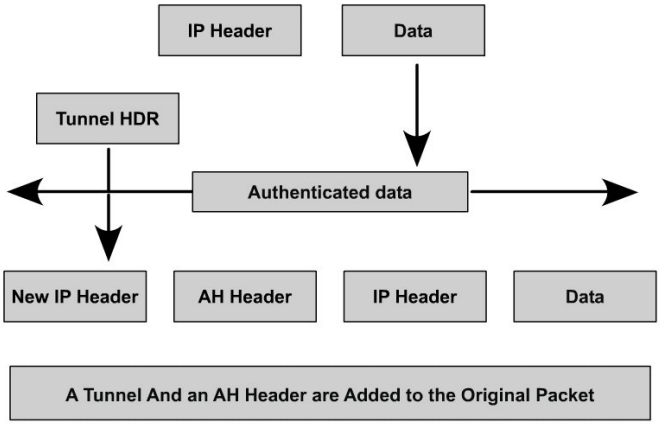
46

AH mode transport



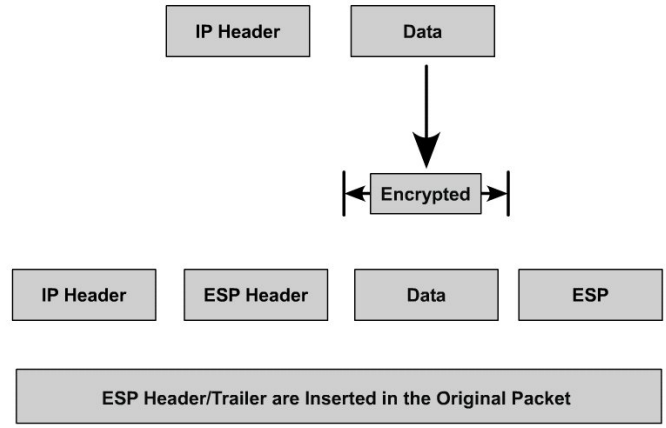
47

AH mode tunnel



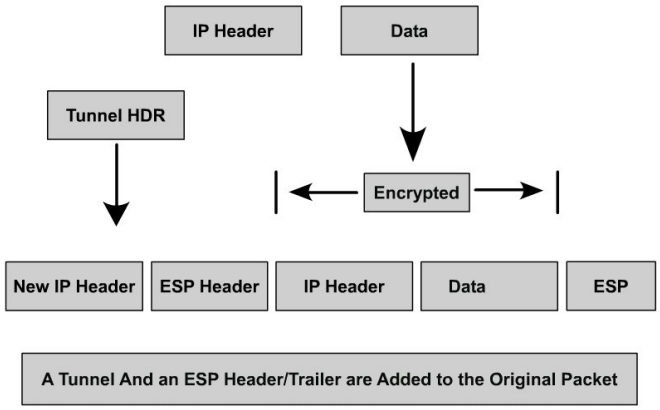
48

ESP mode transport



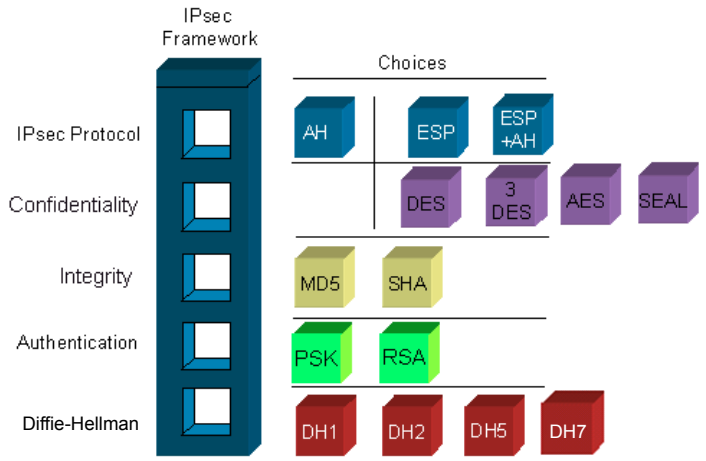
49

ESP mode tunnel

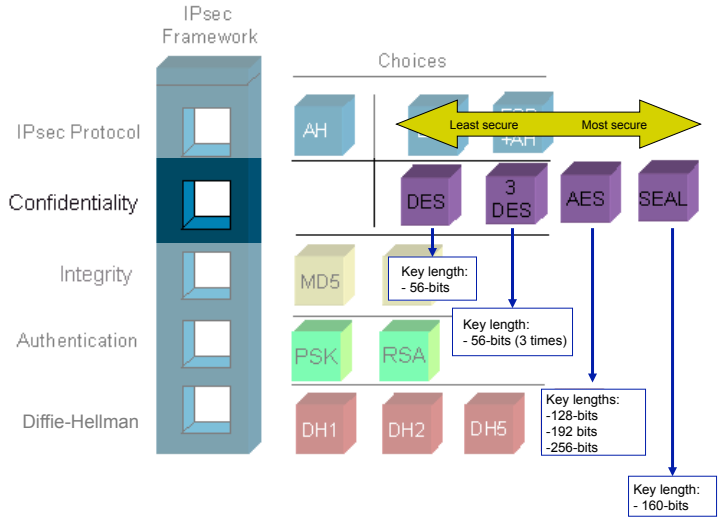


50

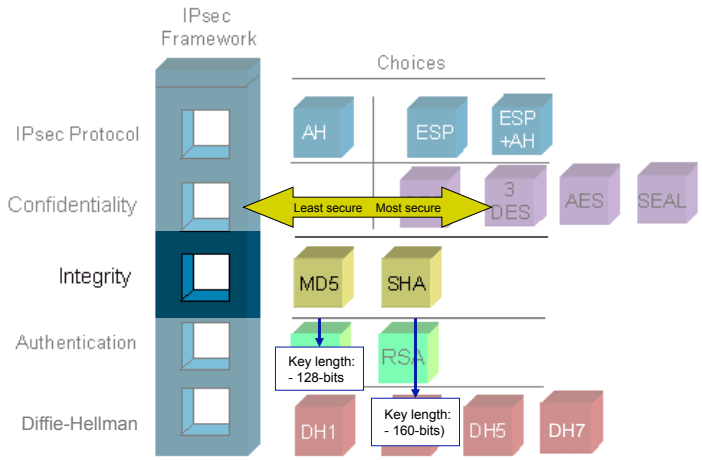
IPSec Framework



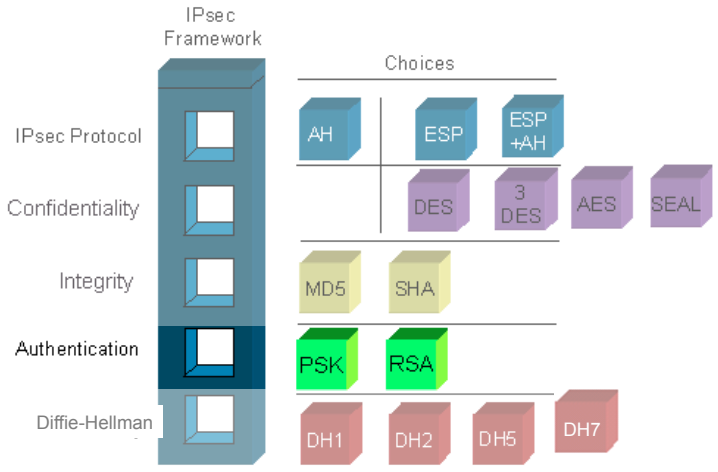
Confidentiality



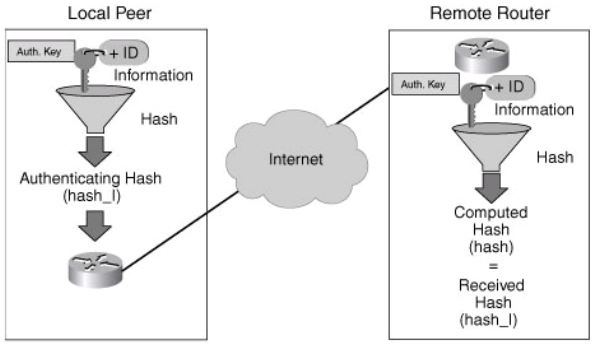
Integrity



Authentication



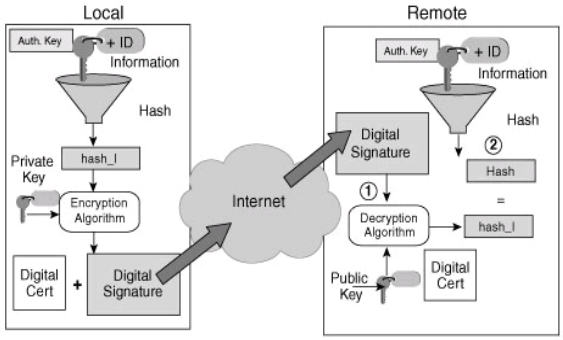
Pre-shared Key (PSK)



•At the local device, the authentication key and the identity information (device-specific information) are sent through a hash algorithm to form hash_I. One-way authentication is established by sending hash_I to the remote device. If the remote device can independently create the same hash, the local device is authenticated.

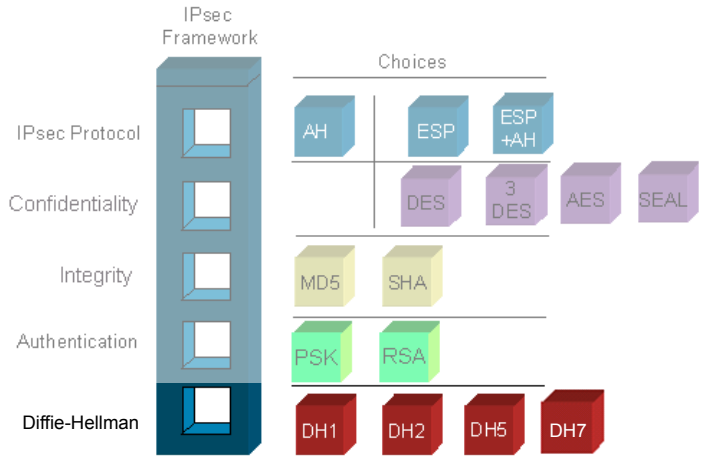
• The authentication process continues in the opposite direction. The remote device combines its identity information with the preshared-based authentication key and sends it through the hash algorithm to form hash_R. hash_R is sent to the local device. If the local device can independently create the same hash, the remote device is authenticated.

RSA Signatures



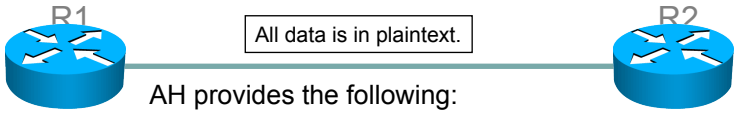
- At the local device, the authentication key and identity information (device-specific information) are sent through the hash algorithm forming hash_I. hash_I is encrypted using the local device's private encryption key creating a digital signature. The digital signature and a digital certificate are forwarded to the remote device. 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_I.
- Next, the remote device independently creates hash_I from stored information. If the calculated hash_I equals the decrypted hash_I, the local device is authenticated. After the remote device authenticates the local device, the authentication process begins in the opposite direction and all steps are repeated from the remote device to the local device.

Secure Key Exchange



IPSec Framework Protocols

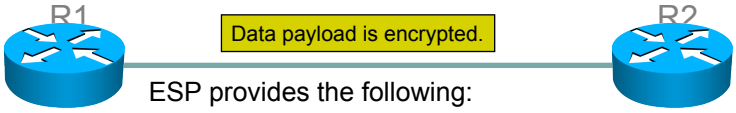
Authentication Header



AH provides the following:

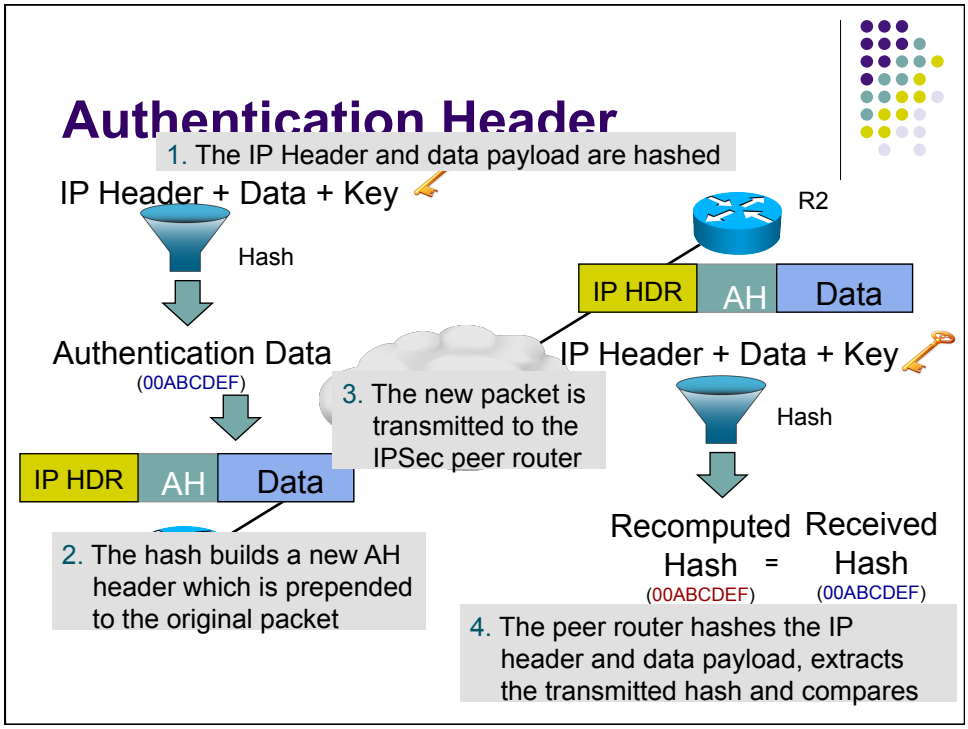
- Authentication
- Integrity

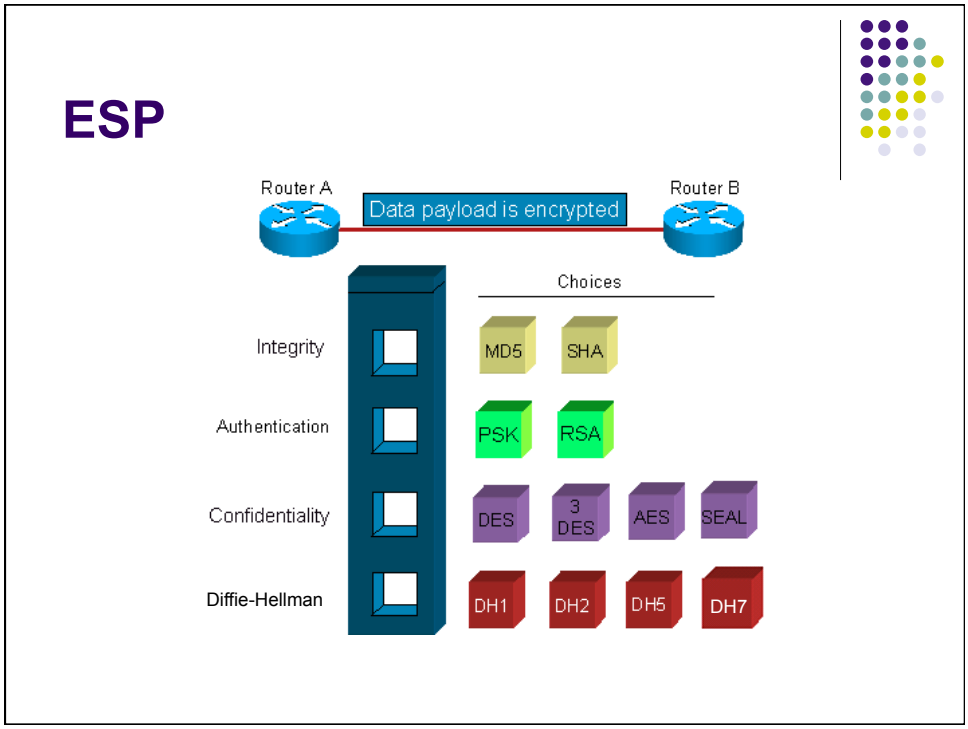
Encapsulating Security Payload

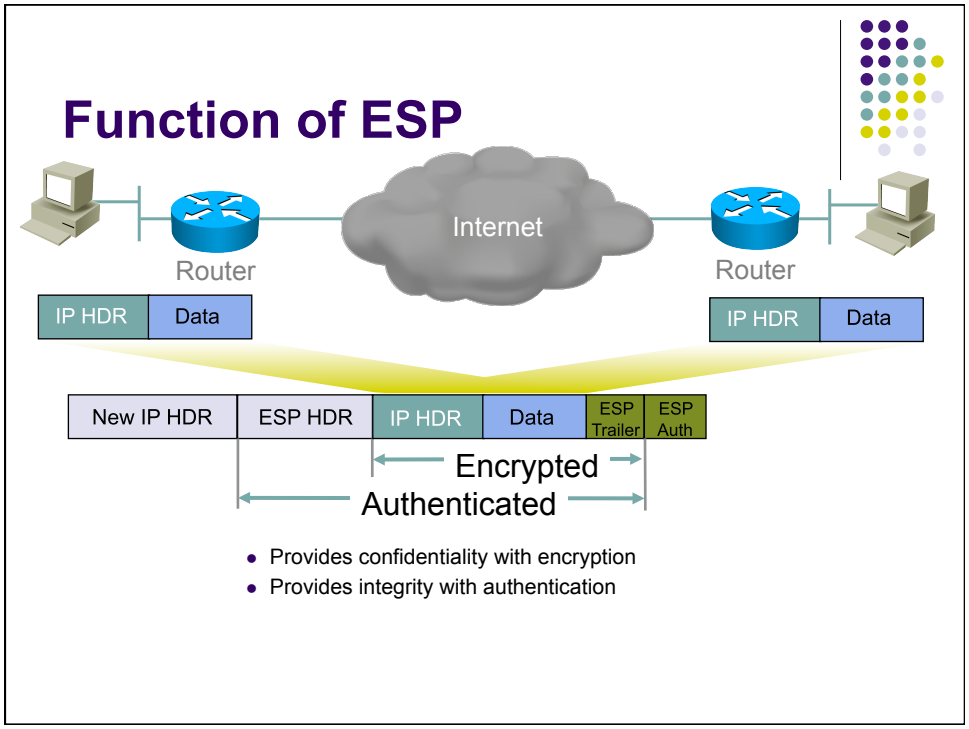


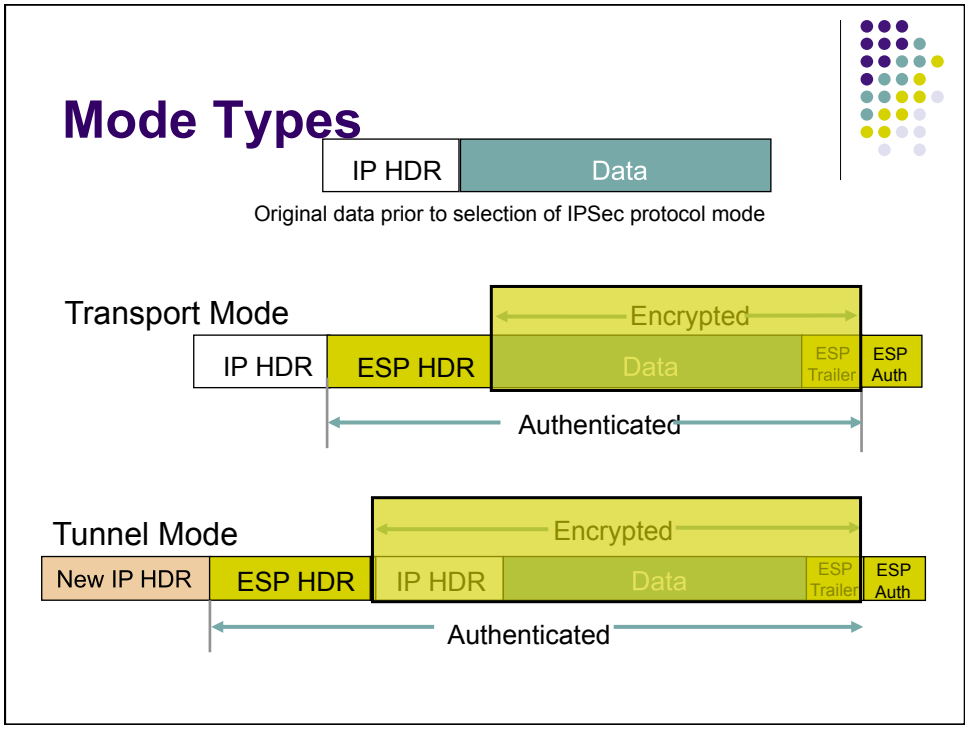
ESP provides the following:

- Encryption
- Authentication
- Integrity

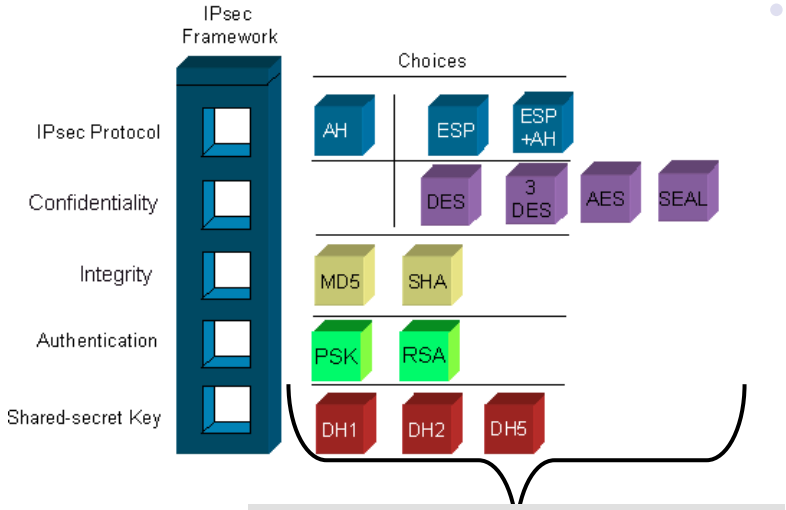

Authentication Header
1. The IP Header and data payload are hashed
IP Header + Data + Key
Hash
Authentication Data
(00ABCDEF)
IP HDR
AH
Data
2. The hash builds a new AH header which is prepended to the original packet
3. The new packet is transmitted to the IPSec peer router
R2
IP HDR
AH
Data
IP Header + Data + Key
Hash
Recomputed Hash
(00ABCDEF)
=
Received Hash
(00ABCDEF)
4. The peer router hashes the IP header and data payload, extracts the transmitted hash and compares


ESP
Router A
Data payload is encrypted
Router B
Choices
Integrity
MD5
SHA
Authentication
PSK
RSA
Confidentiality
DES
3 DES
AES
SEAL
Diffie-Hellman
DH1
DH2
DH5
DH7


Function of ESP
Internet
Router
Router
IP HDR
Data
IP HDR
Data
New IP HDR
ESP HDR
IP HDR
Data
ESP Trailer
ESP Auth
Encrypted
Authenticated
Provides confidentiality with encryption
Provides integrity with authentication

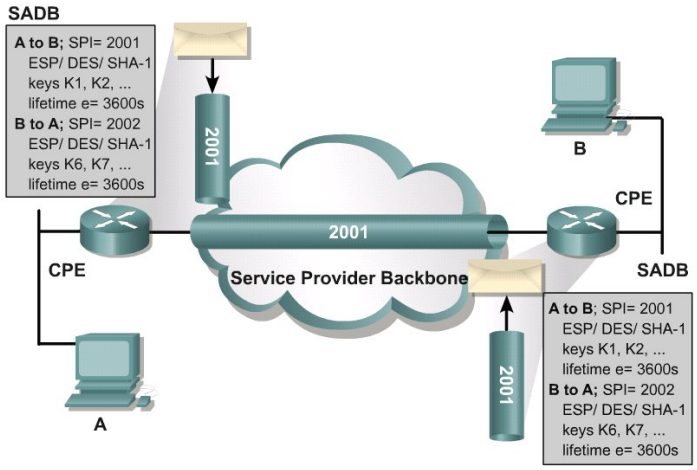

Mode Types
IP HDR
Data
Original data prior to selection of IPSec protocol mode
Transport Mode
Encrypted
IP HDR
ESP HDR
Data
ESP Trailer
ESP Auth
Authenticated
Tunnel Mode
Encrypted
New IP HDR
ESP HDR
IP HDR
Data
ESP Trailer
ESP Auth
Authenticated

Securitv Associations



IPSec parameters are configured using IKE

Security associations



64

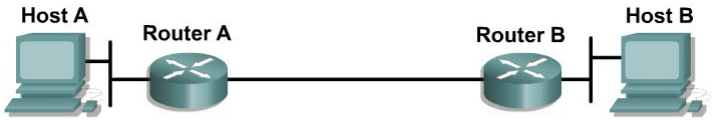
Security associations

An SA contains the following security parameters

- Authentication/encryption algorithm, key length and other encryption parameters, such as key lifetime, used with protected packets
- Session keys for authentication, or HMACs, and encryption, which can be entered manually or negotiated automatically with the help of the IKE protocol, fed to the algorithms
- A specification of network traffic to which the SA will be applied, such as all IP traffic or only TELNET sessions
- IPSec AH or ESP encapsulation protocol and tunnel or transport mode

65

Étapes IPSEC



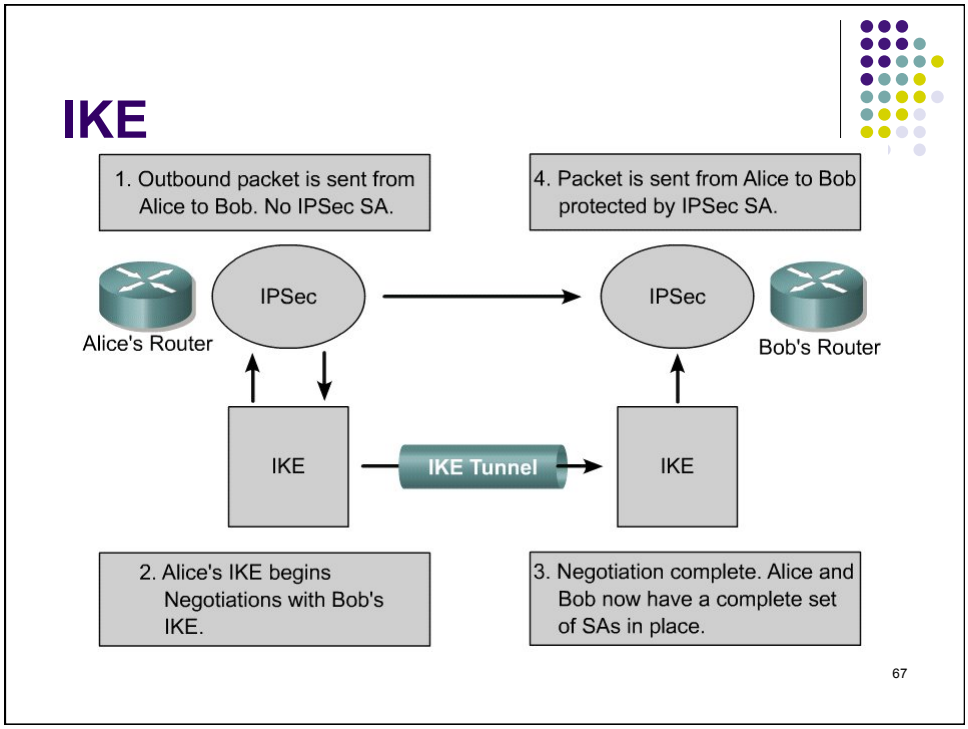
1. Host A sends interesting traffic to Host B.
2. Router A and B negotiate an IKE Phase One session.

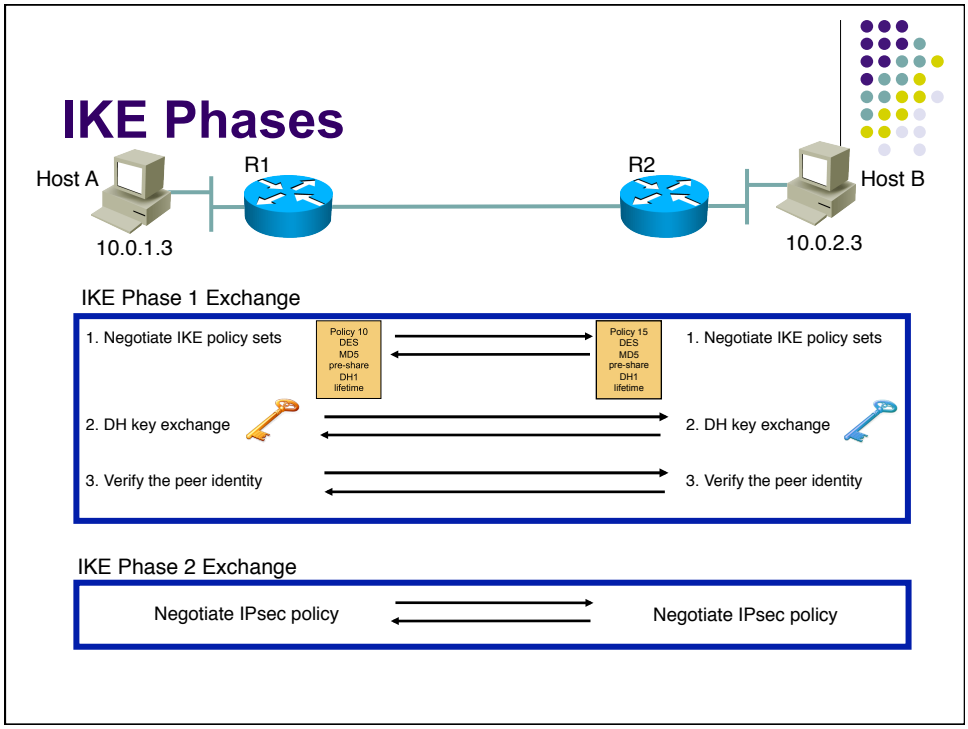
3. Router A and B negotiate an IKE Phase Two session.

4. Information is exchanged via IPSec tunnel.

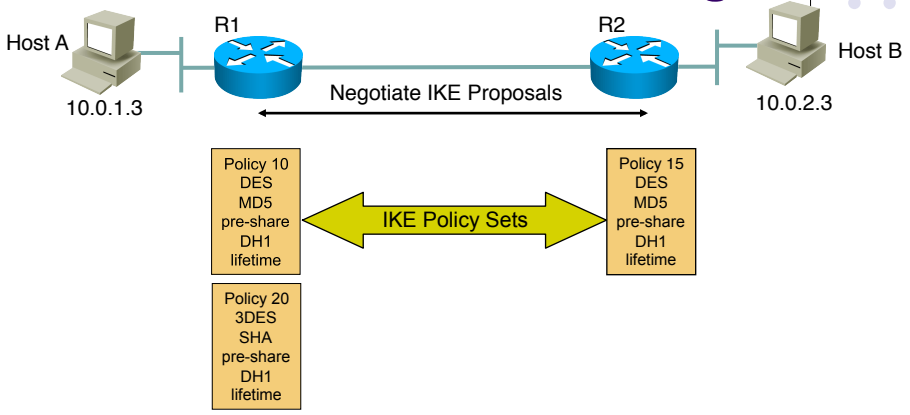
5. IPSec tunnel is terminated.

66


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.
IPSec
IPSec
Alice's Router
Bob's Router
IKE
IKE Tunnel
IKE
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.
67


IKE Phases
Host A
R1
R2
Host B
10.0.1.3
10.0.2.3
IKE Phase 1 Exchange
1. Negotiate IKE policy sets
Policy 10
DES
MD5
pre-share
DH1
lifetime
Policy 15
DES
MD5
pre-share
DH1
lifetime
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

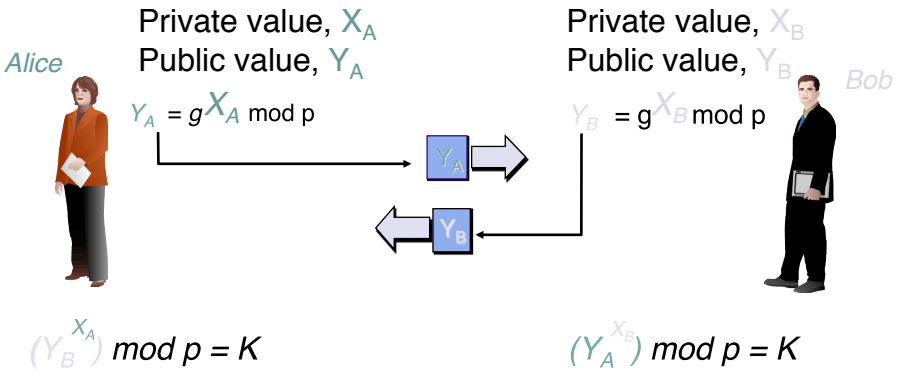
IKE Phase 1 – First Exchange



Negotiates matching IKE policies to protect IKE exchange

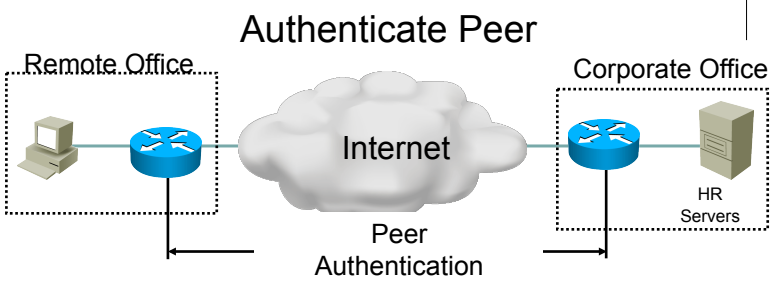
IKE Phase 1 – Second Exchange

Establish DH Key



A DH exchange is performed to establish keying material.

IKE Phase 1 – Third Exchange



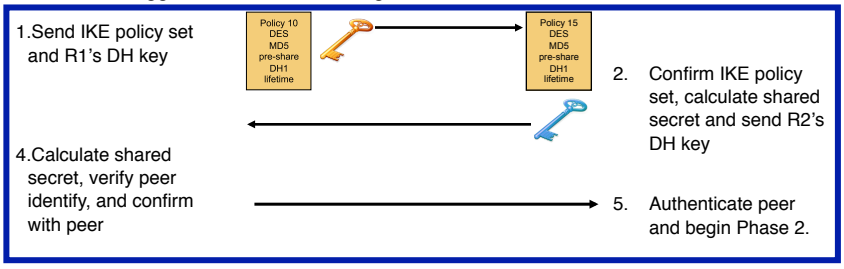
Peer authentication methods

- PSKs
- RSA signatures
- RSA encrypted nonces

A bidirectional IKE SA is now established.

IKE Phase 1 – Aggressive Mode

IKE Phase 1 Aggressive Mode Exchange



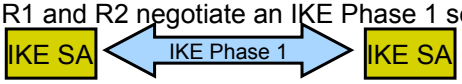
IKE Phase 2 Exchange

IKE Phase 2

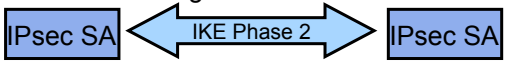
- IKE negotiates matching IPsec policies.
- Upon completion, unidirectional IPsec Security Associations(SA) are established for each protocol and algorithm combination.

IPSec VPN Negotiation

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.

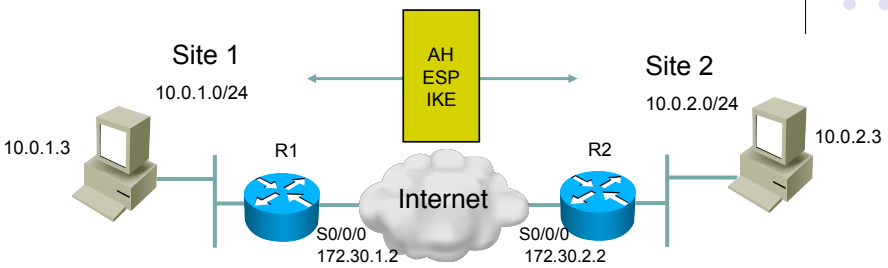
5. The IPsec tunnel is terminated.

Configuring IPsec

Tasks to Configure IPsec:

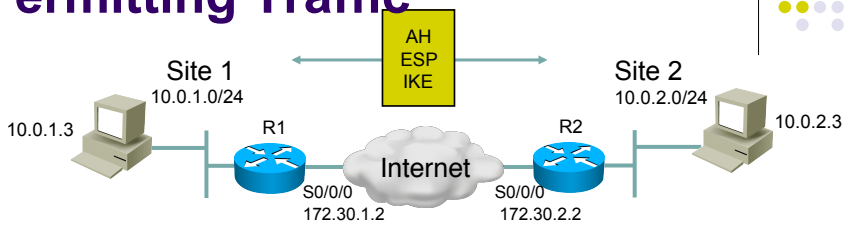
Task 1: Ensure that ACLs are compatible with IPsec.
Task 2: Create ISAKMP (IKE) policy.
Task 3: Configure IPsec transform set.
Task 4: Create a crypto ACL.
Task 5: Create and apply the crypto map.

Task 1
Configure Compatible ACLs



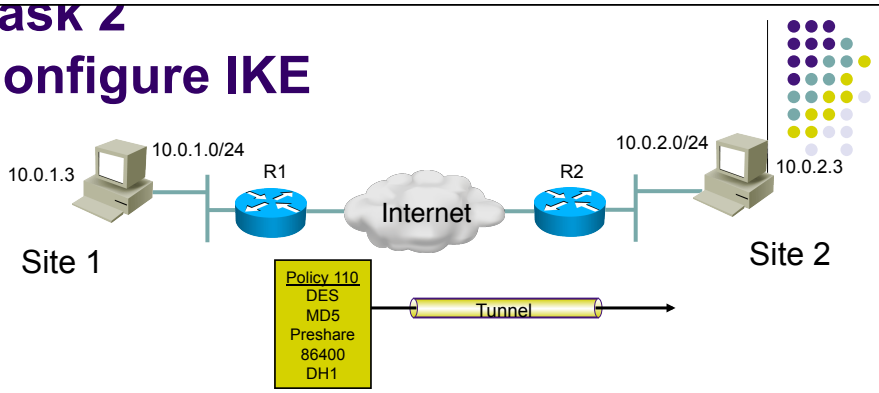
- Ensure that protocols 50 (ESP), 51 (AH) and UDP port 500 (ISAKMP) traffic are not blocked by incoming ACLs on interfaces used by IPsec.

Permitting Traffic




```
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#
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
R1#
```

Task 2
Configure IKE




```
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 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

Multiple Policies

R1(config)#

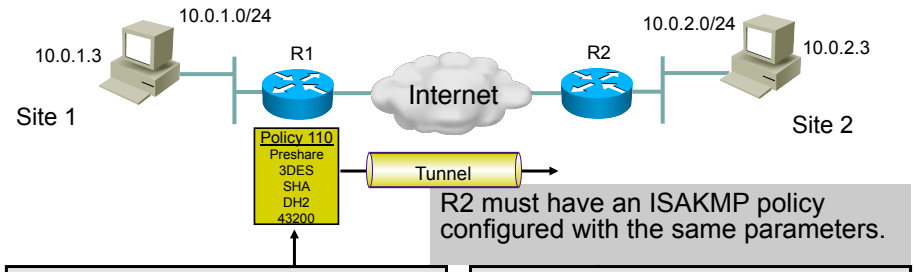
```
crypto isakmp policy 100
  hash md5
  authentication pre-share
!
crypto isakmp policy 200
  hash sha
  authentication rsa-sig
!
crypto isakmp policy 300
  hash md5
  authentication pre-share
```

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
```

Policy Negotiations

R1 attempts to establish a VPN tunnel with R2 and sends its IKE policy parameters



R2 must have an ISAKMP policy configured with the same parameters.

```
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
```

Crypto ISAKMP Key

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

Sample Configuration


```
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 cisco1234 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)#
```

Task 3
Configure the Transform Set

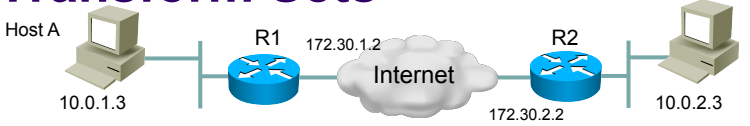
```
router(config)#
crypto ipsec transform-set transform-set-name
   transform1 [transform2] [transform3]]
```

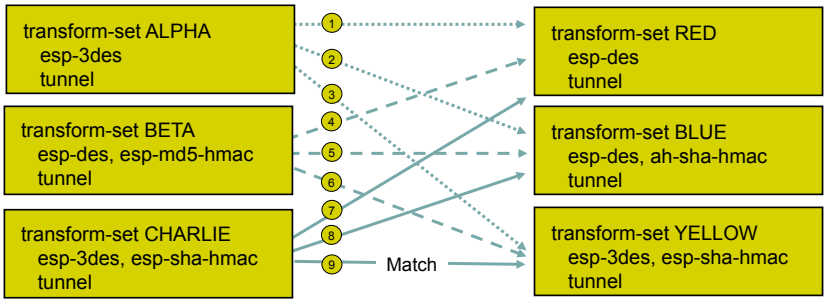
crypto ipsec transform-set Parameters

Command	Description
transform-set-name	This parameter specifies the name of the transform set to create (or modify).
transform1, *transform2*, *transform3*	Type of transform set. You may 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.

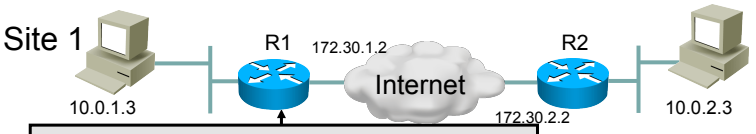
Transform Sets





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

Sample Configuration



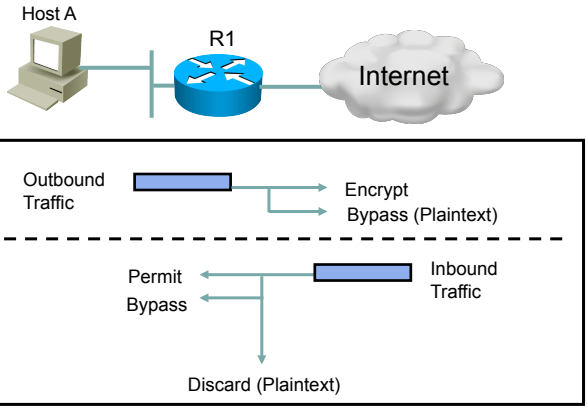

```
R1(config)# crypto isakmp key cisco123 address 172.30.2.2
R1(config)# crypto ipsec transform-set MYSET esp-aes 128
R1(cfg-crypto-trans)# exit
R1(config)#
```

Note:

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

```
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
```

Task 4
Configure the Crypto ACLs



- Outbound indicates the data flow to be protected by IPsec.
- Inbound filters and discards traffic that should have been protected by IPsec.

Command Syntax


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

access-list access-list-number Parameters

access-list *access-list-number* **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 traffic IP traffic that matches 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.

Symmetric Crypto ACLs


```
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

(when evaluating inbound traffic- source: 10.0.2.0, destination: 10.0.1.0)
```

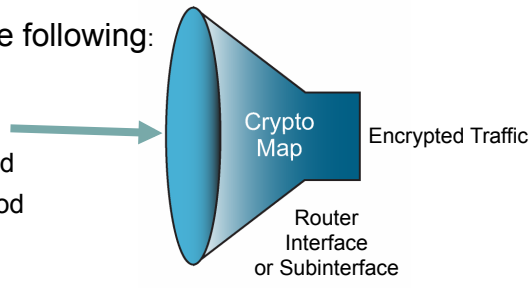
```
Applied to R2 S0/0/0 outbound traffic:
R2(config)# access-list 101 permit tcp 10.0.2.0 0.0.0.255 10.0.1.0 0.0.0.255

(when evaluating inbound traffic- source: 10.0.1.0, destination: 10.0.2.0)
```

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



Crypto Map Command

router(config)#

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

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

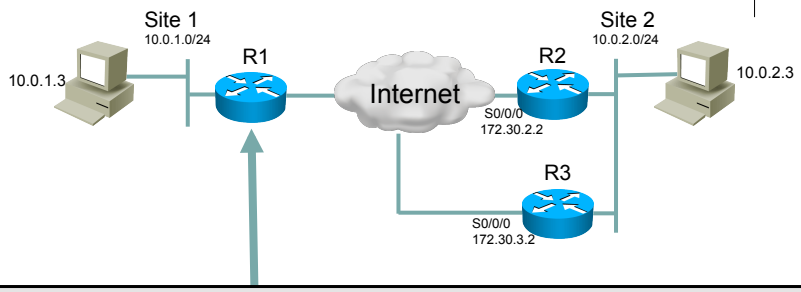
crypto map Parameters

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.

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.

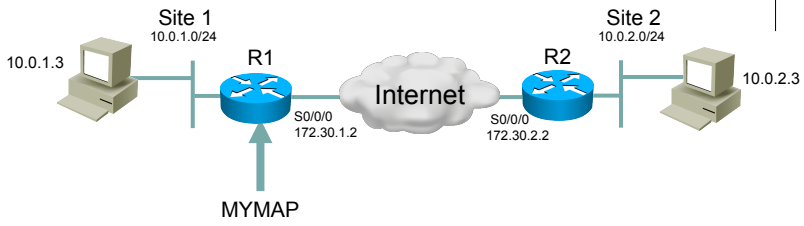
Sample Configuration




```
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.

Assign the Crypto Map Set




```
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

CLI Commands

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

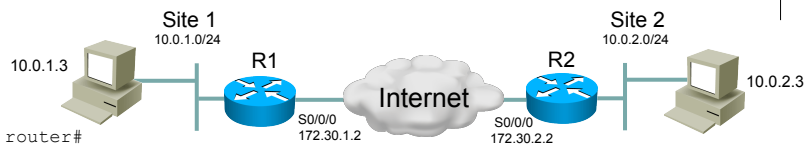
show crypto map


```
router#
show crypto map
```

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, }
```

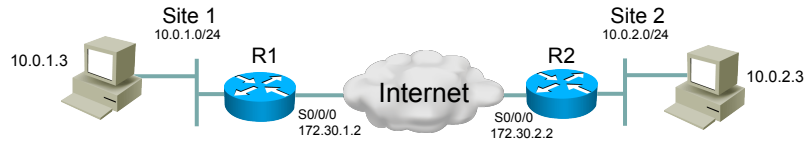
show crypto isakmp policy




```
show crypto isakmp policy
```

```
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
```

show crypto ipsec transform-set

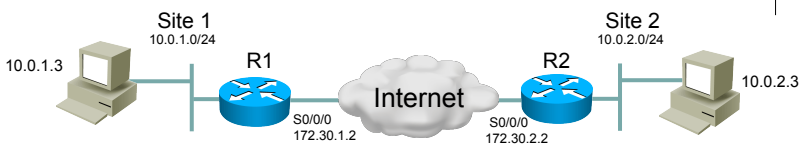



```
show crypto ipsec transform-set
```

Displays the currently defined transform sets

```
R1# show crypto ipsec transform-set
Transform set AES_SHA: { esp-128-aes esp-sha-hmac }
      will negotiate = { Tunnel,  },
```

show crypto ipsec sa




```
R1# show crypto ipsec sa
Interface: Serial0/0/0
        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, flacs={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: 0AB1C9C
```

debug crypto isakmp

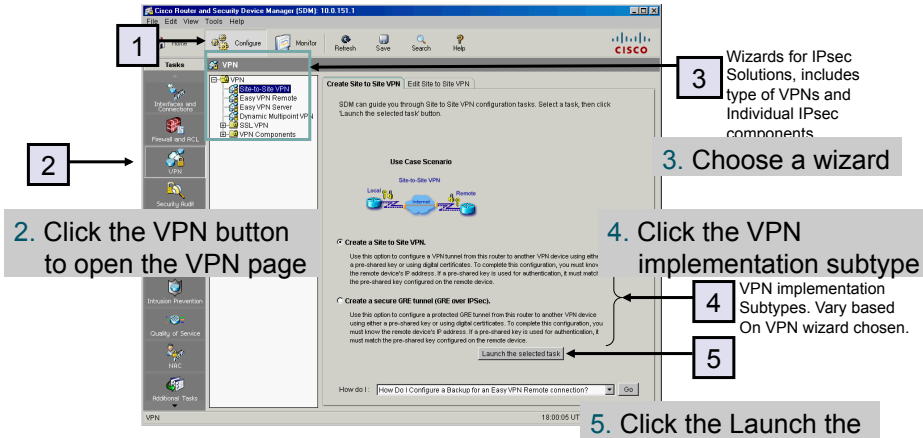
```
router#
debug crypto isakmp
```

```
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
```

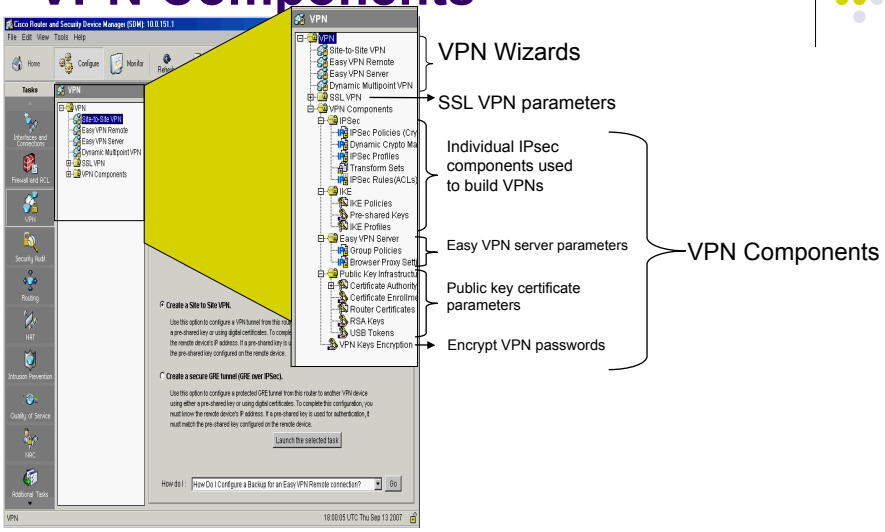
- This is an example of the Main Mode error message.
- The failure of Main Mode suggests that the Phase I policy does not match on both sides.
- Verify that the Phase I policy is on both peers and ensure that all the attributes match.

Starting a VPN Wizard

1. Click Configure in main toolbar

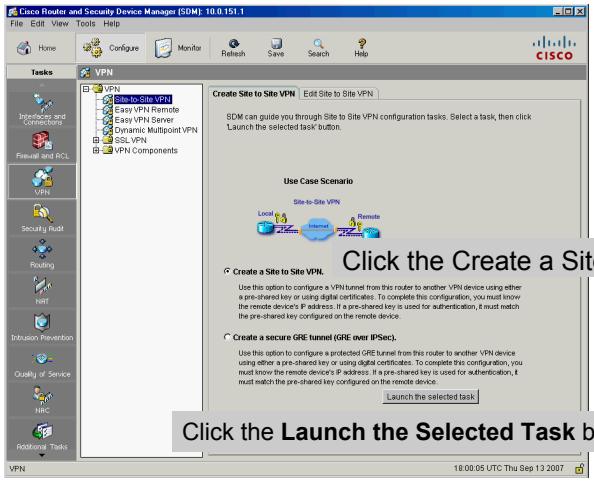


VPN Components



Configuring a Site-to-Site VPN

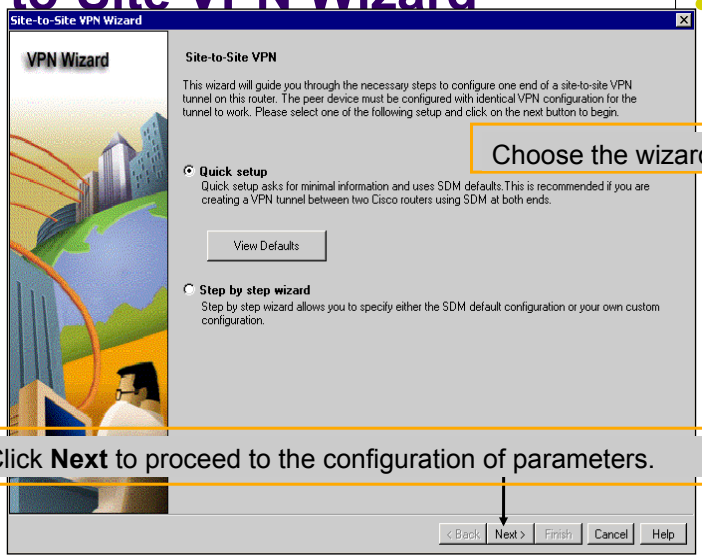
Choose Configure > VPN > Site-to-Site VPN



Click the Create a Site-to-Site VPN

Click the **Launch the Selected Task** button

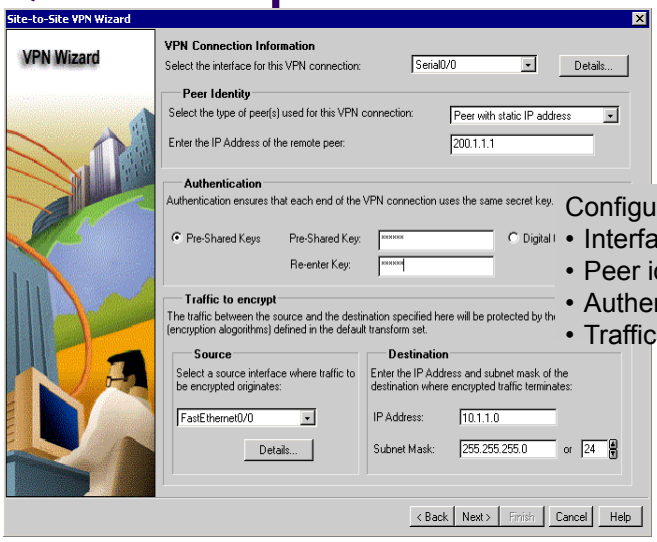
Site-to-Site VPN Wizard



Choose the wizard mode

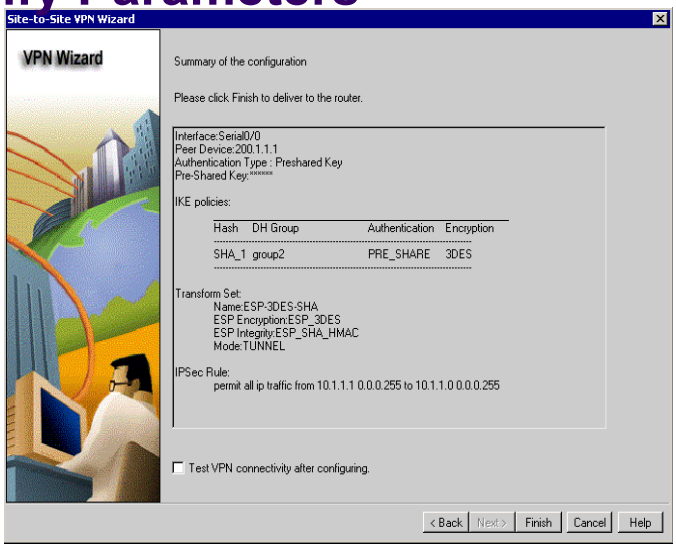
Click **Next** to proceed to the configuration of parameters.

Quick Setup

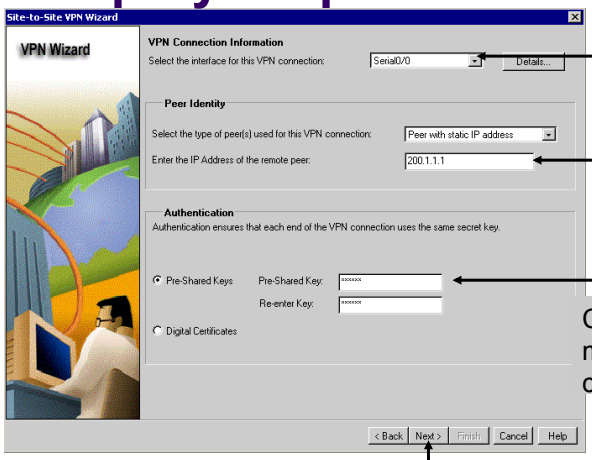


Configure the parameters
- Interface to use
- Peer identity information
- Authentication method
- Traffic to encrypt

Verify Parameters



Step-by-Step Wizard



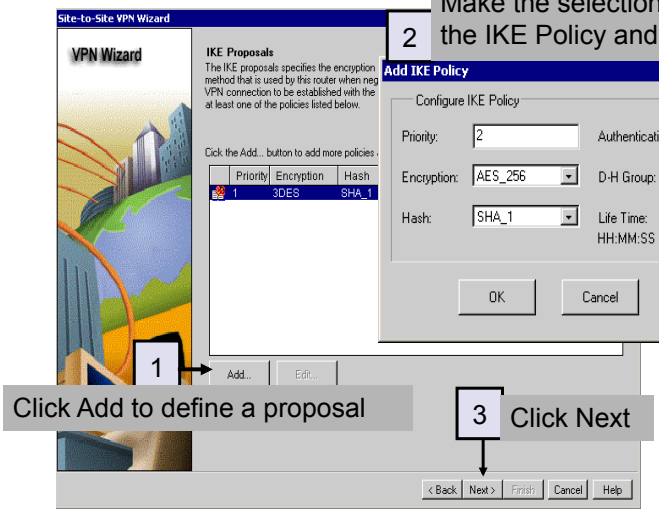
1 Choose the outside interface that is used to connect to the IPSec peer

2 Specify the IP address of the peer

3 Choose the authentication method and specify the credentials

4 Click Next

Creating a Custom IKE Proposal



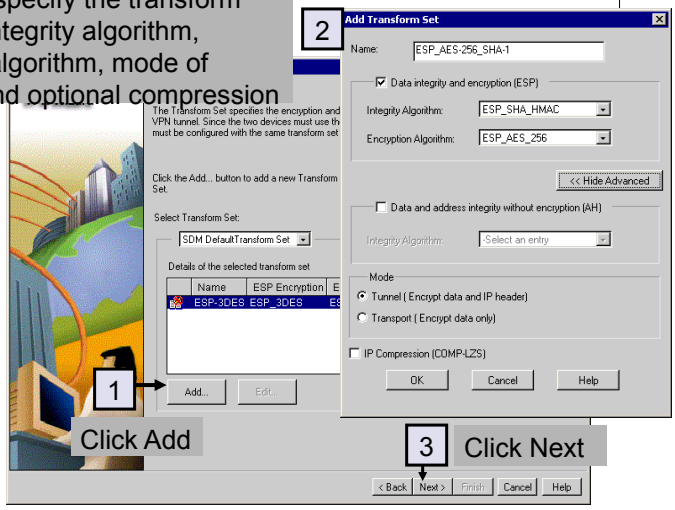
2 Make the selections to configure the IKE Policy and click OK

1 Click Add to define a proposal

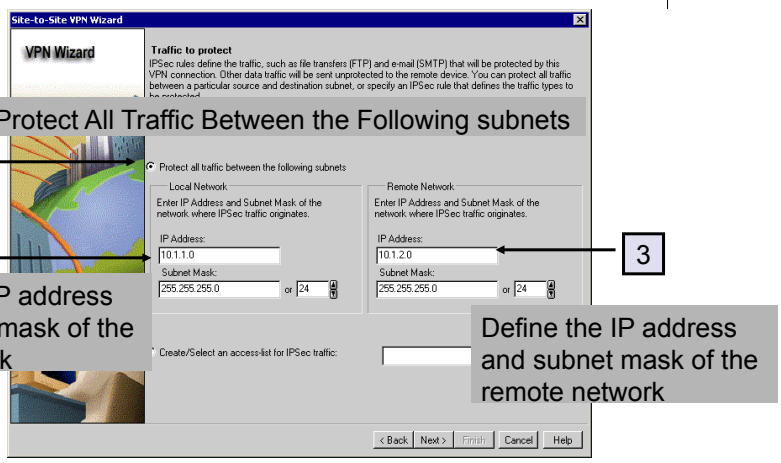
3 Click Next

Creating a Custom IPSec Transform Set

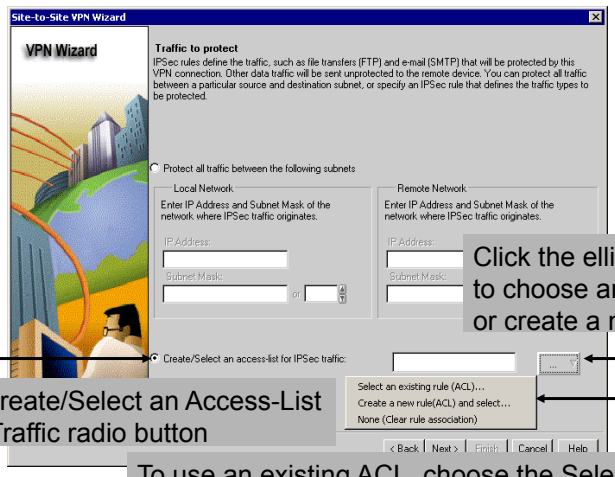
Define and specify the transform set name, integrity algorithm, encryption algorithm, mode of operation and optional compression



Protecting Traffic Subnet to Subnet



Protecting Traffic Custom ACL



1

Click the Create/Select an Access-List for IPSec Traffic radio button

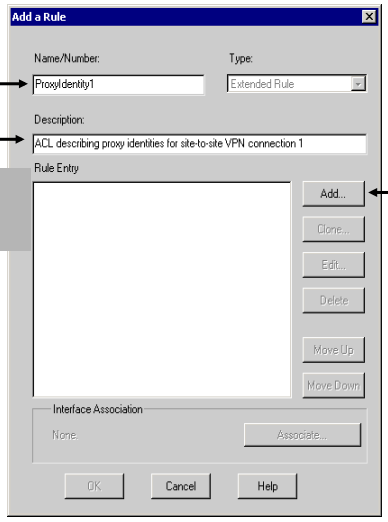
Click the ellipses button to choose an existing ACL or create a new one

2

3

To use an existing ACL, choose the Select an Existing Rule (ACL) option. To create a new ACL, choose the Create a New Rule (ACL) and Select option

Add a Rule



1

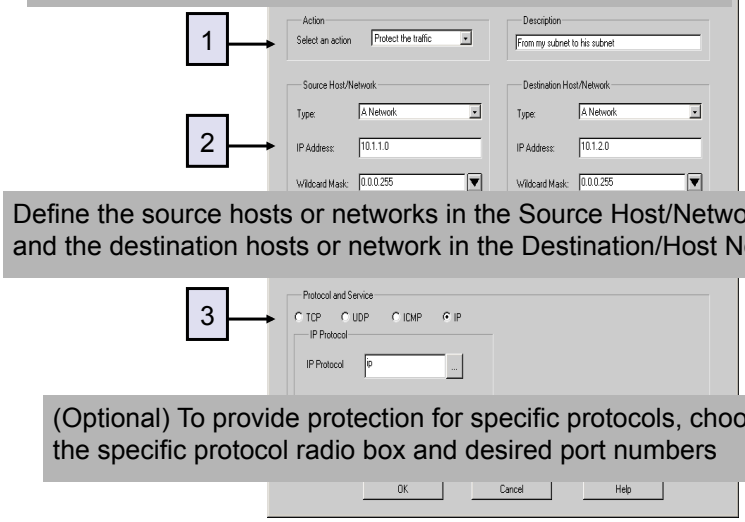
Give the access rule a name and description

2

Click Add

Configuring a New Rule Entry

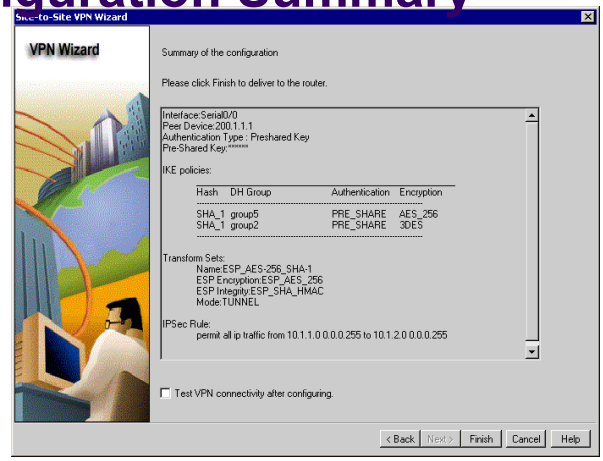
Choose an action and enter a description of the rule entry



Define the source hosts or networks in the Source Host/Network pane and the destination hosts or network in the Destination/Host Network pane

(Optional) To provide protection for specific protocols, choose the specific protocol radio box and desired port numbers

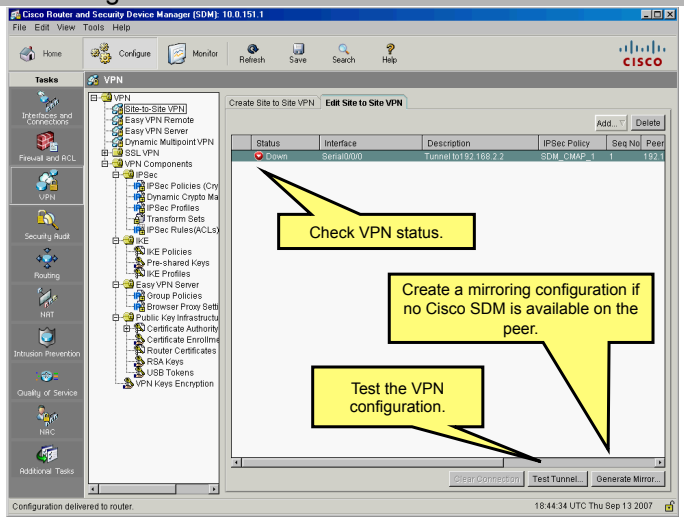
Configuration Summary



- Click **Back** to modify the configuration.
- Click **Finish** to complete the configuration.

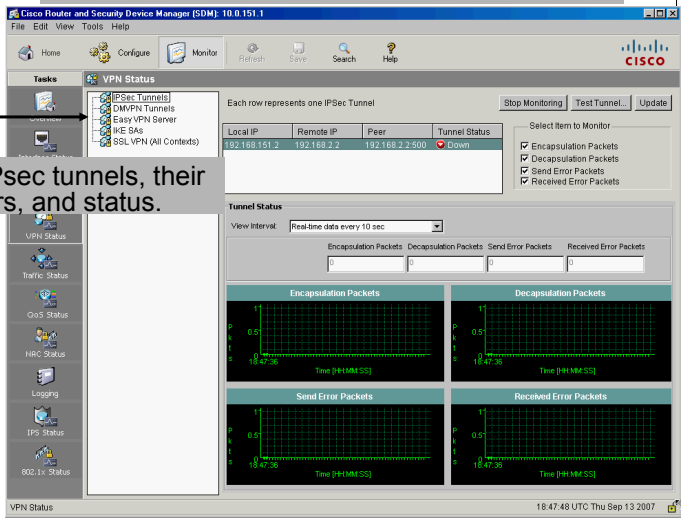
Verify VPN Configuration

Choose Configure > VPN > Site-to-Site VPN > Edit Site-to-Site VPN



Monitor

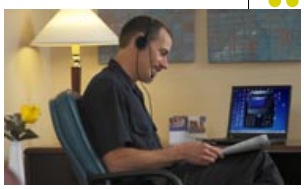
Choose Monitor > VPN Status > IPSec Tunnels



Lists all IPsec tunnels, their parameters, and status.

Telecommuting

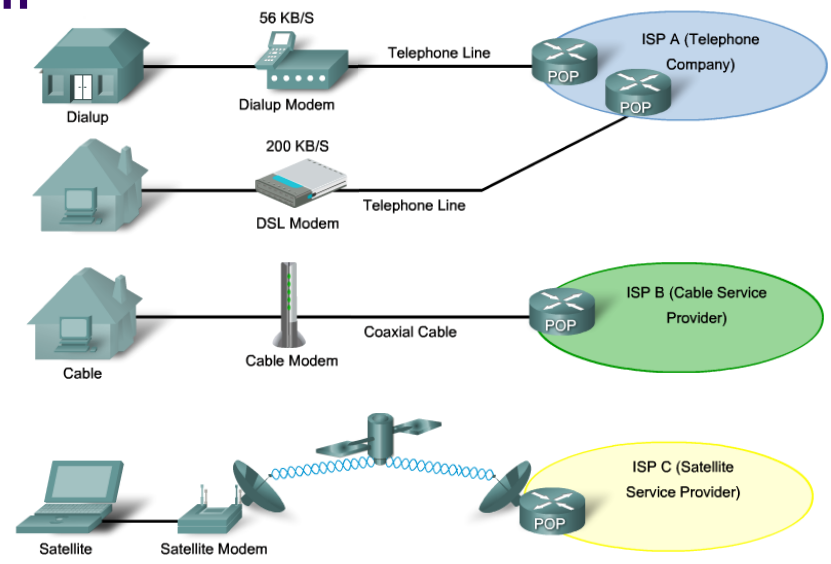
- Flexibility in working location and working hours
- Employers save on real-estate, utility and other overhead costs
- Succeeds if program is voluntary, subject to management discretion, and operationally feasible



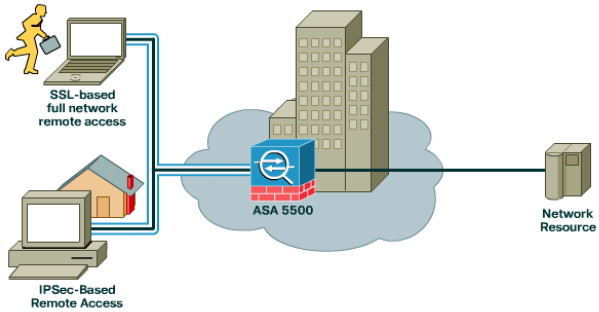
Telecommuting Benefits

- Organizational benefits:
  - Continuity of operations
  - Increased responsiveness
  - Secure, reliable, and manageable access to information
  - Cost-effective integration of data, voice, video, and applications
  - Increased employee productivity, satisfaction, and retention
- Social benefits:
  - Increased employment opportunities for marginalized groups
  - Less travel and commuter related stress
- Environmental benefits:
  - Reduced carbon footprints, both for individual workers and organizations

Implementing Remote Access



Methods for Deploying Remote Access



IPsec Remote Access VPN

Any Application ⟷ Anywhere Access

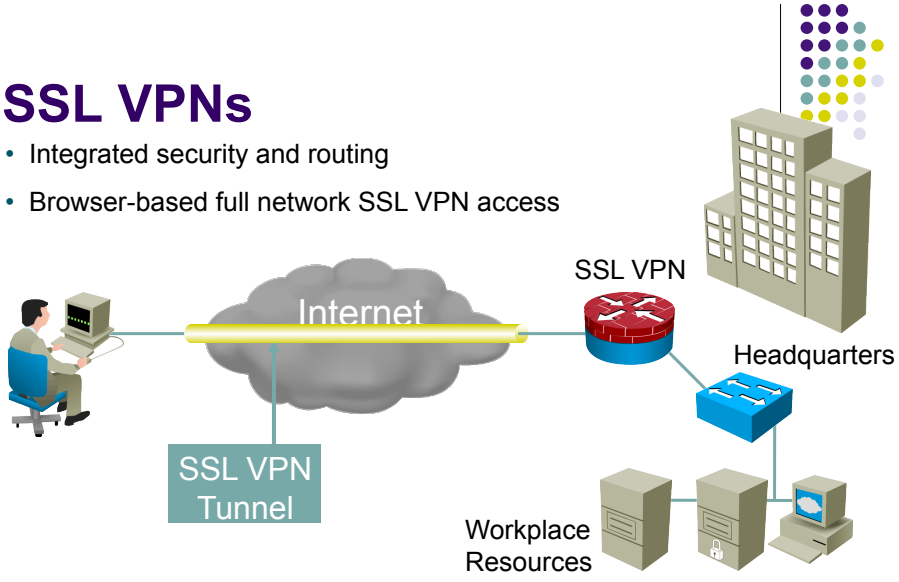
SSL-Based VPN

Comparison of SSL and IPSec

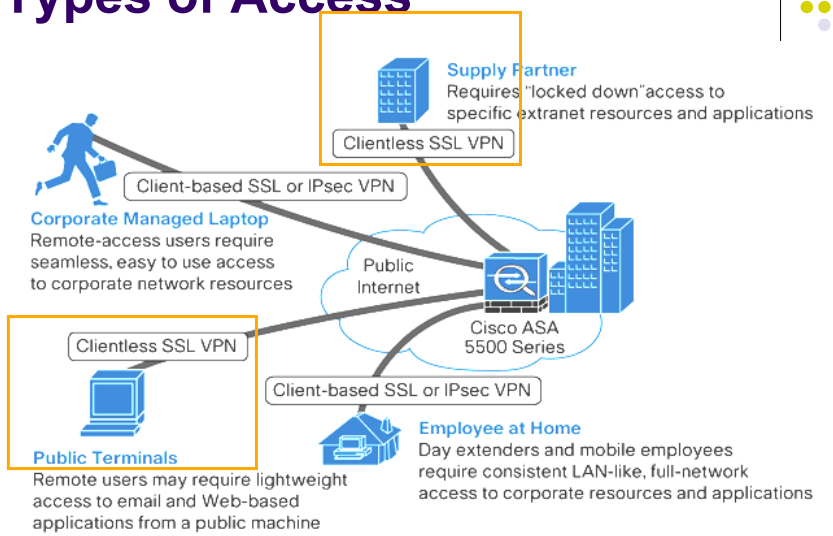
	SSL	IPsec
Applications	Web-enabled applications, file sharing, e-mail	All IP-based applications
Encryption	**Moderate** Key lengths from 40 bits to 128 bits	**Stronger** Key lengths from 56 bits to 256 bits
Authentication	**Moderate** One-way or two-way authentication	**Strong** Two-way authentication using shared secrets or digital certificates
Ease of Use	**Very high**	**Moderate** Can be challenging to nontechnical users
Overall Security	**Moderate** Any device can connect	**Strong** Only specific devices with specific configurations can connect

SSL VPNs

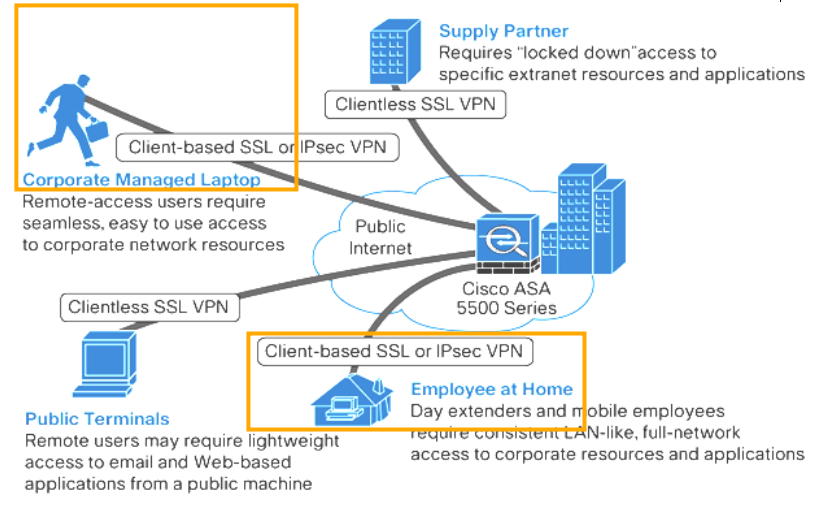
- Integrated security and routing
- Browser-based full network SSL VPN access



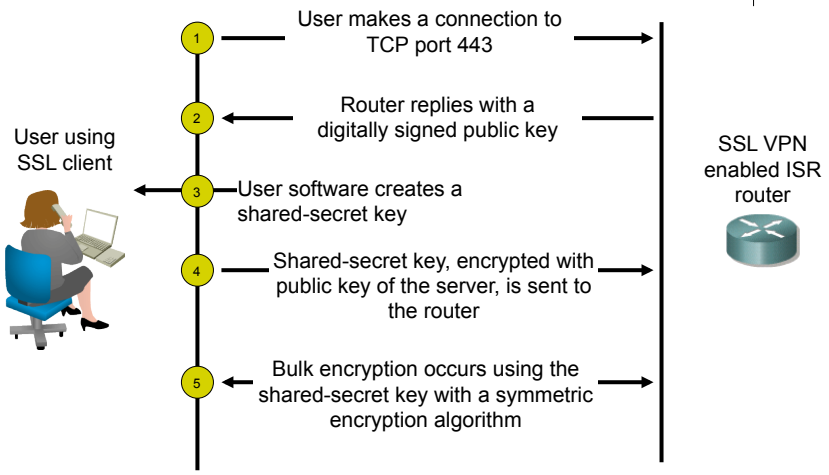
Types of Access



Full Tunnel Client Access Mode

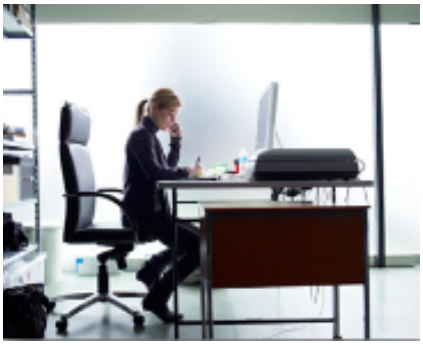


Establishing an SSL Session



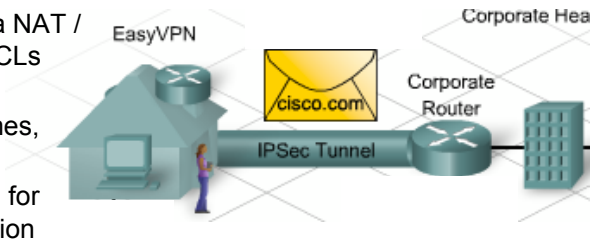
SSL VPN Design Considerations

- User connectivity
- Router feature
- Infrastructure planning
- Implementation scope

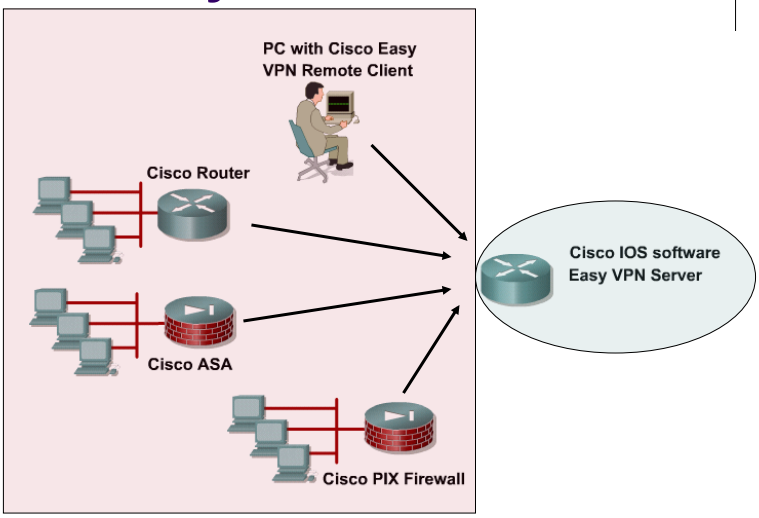


Cisco Easy VPN

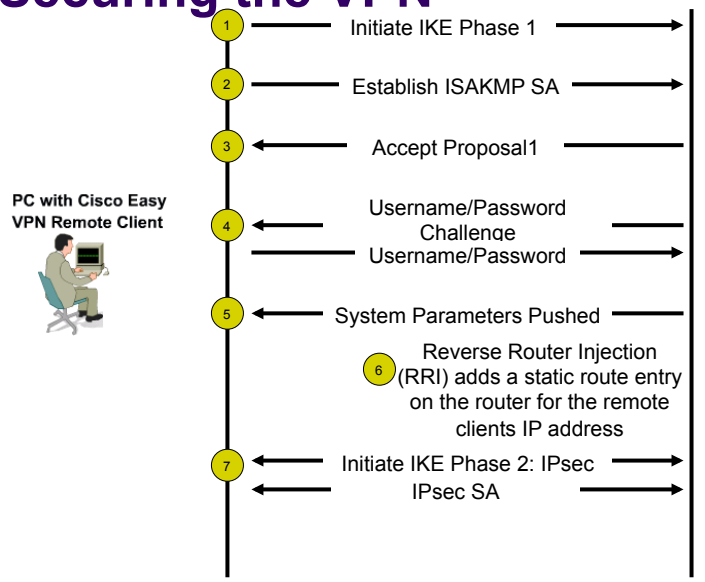
- Negotiates tunnel parameters
- Establishes tunnels according to set parameters
- Automatically creates a NAT / PAT and associated ACLs
- Authenticates users by usernames, group names, and passwords
- Manages security keys for encryption and decryption
- Authenticates, encrypts, and decrypts data through the tunnel



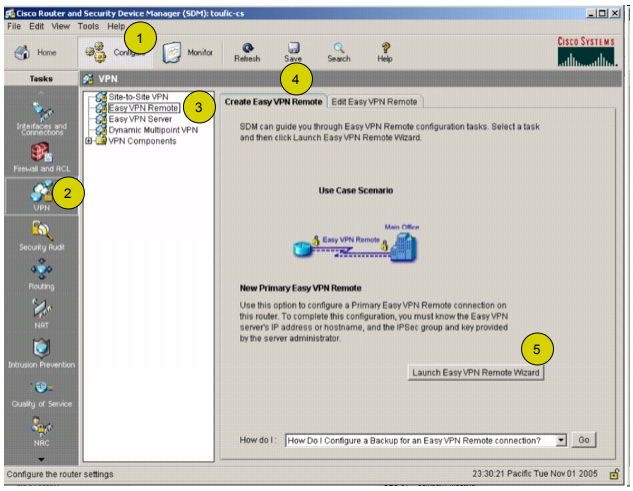
Cisco Easy VPN



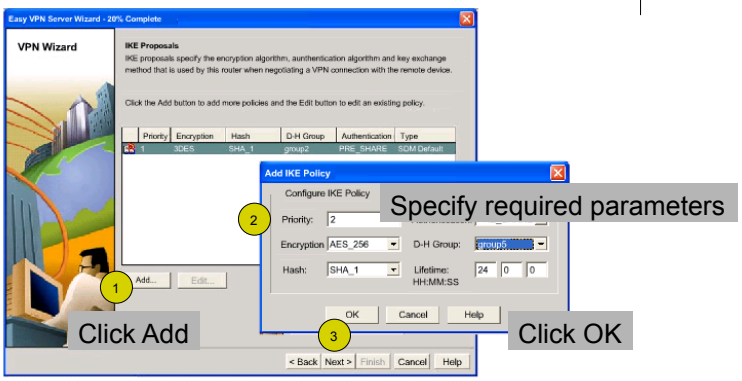
Securing the VPN



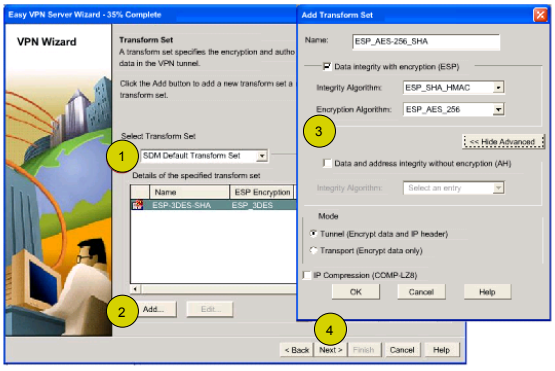
Configuring Cisco Easy VPN Server



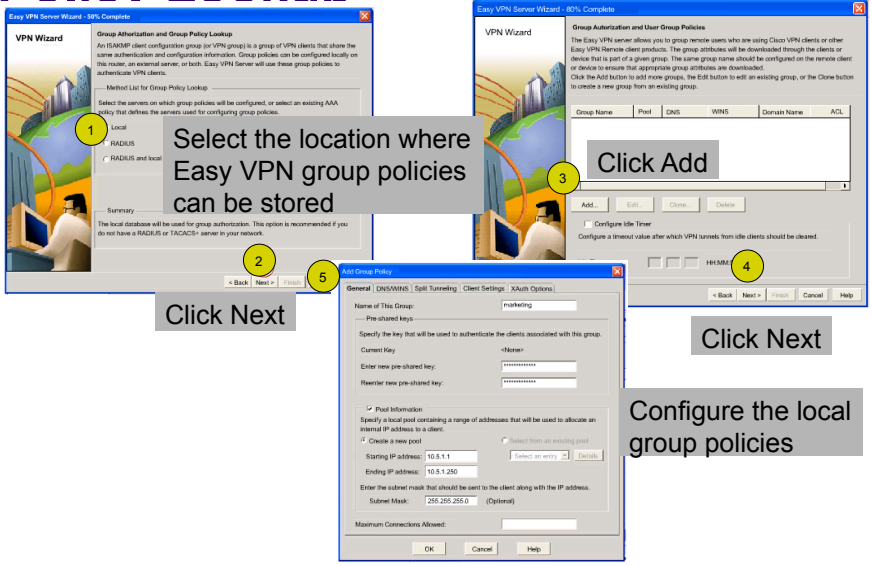
Configuring IKE Proposals



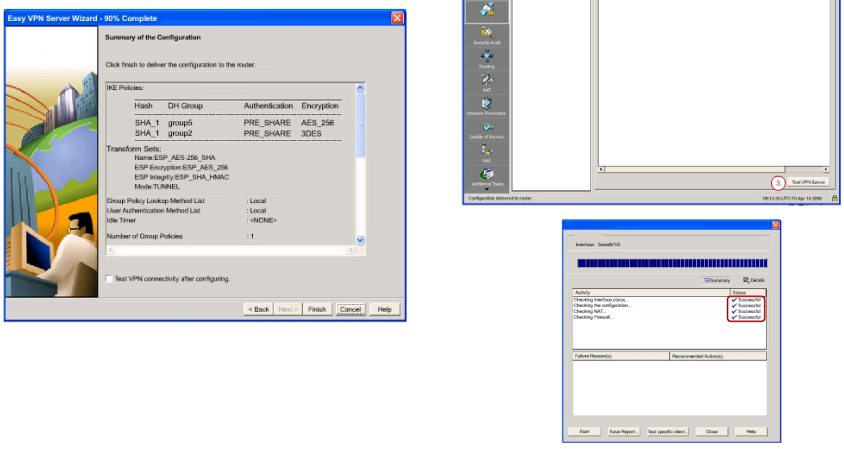
Creating an IPSec Transform Set



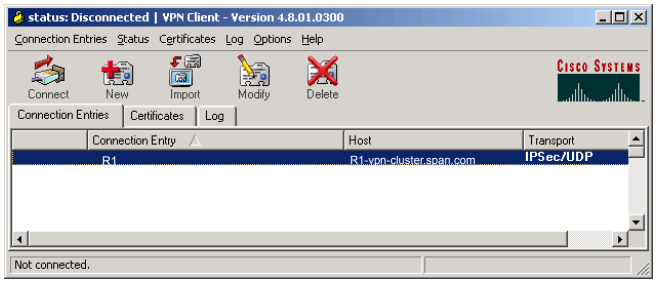
Group Authorization and Group Policy Lookup



Summary of Configuration Parameters



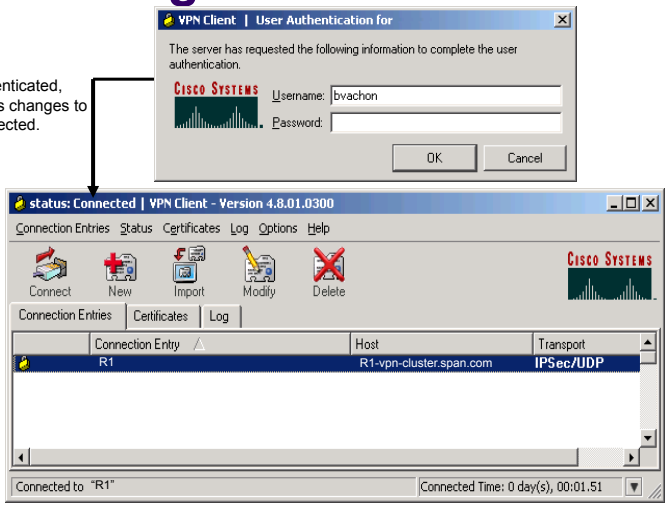
VPN Client Overview



- Establishes end-to-end, encrypted VPN tunnels for secure connectivity
- Compatible with all Cisco VPN products
- Supports the innovative Cisco Easy VPN capabilities

Establishing a Connection

Once authenticated, status changes to connected.



Configuration IPSEC

- http://cisco.netacad.net
- http://www.cisco.com/en/US/tech/tk583/tk372/tech_configuration_examples_list.html
- http://www.vpnc.org/
- Linux :
  - Noyau 2.6 Ipsec intégré+ Racoon
  - Noyau 2.4 FreeS/WAN

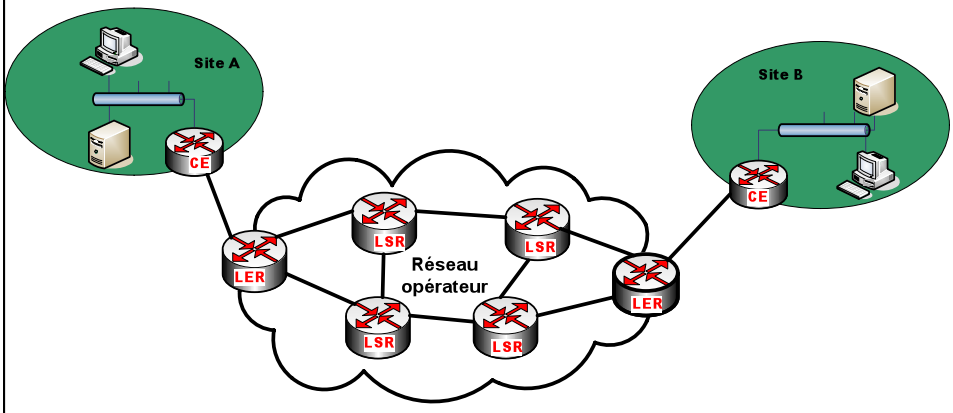
137

VPN MPLS

- Multiprotocol Label Switching
- Gérés uniquement par des opérateurs
- Etanchéité des flux au sens commutation de paquets
- Deux types d'équipements :
  - LER Label Edge Routers
  - LSR Label Switch Routers

138

VPN MPLS



139

VPN MPLS

- Les LER sont des routeurs de périphéries qui marquent le trafic à l'entrée du réseau MPLS.
  - Ils encapsulent les datagrammes d'un protocole spécifiques (par exemple IP) dans les datagrammes MPLS.
  - Cette encapsulation consiste à rajouter une étiquette (label) dépendant de la destination, de la nature et de la priorité du trafic.

140

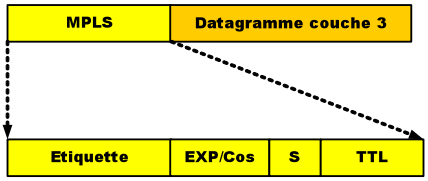
VPN MPLS

- Les LSR analysent les étiquettes des datagrammes MPLS et traient chaque datagramme selon l'information contenue dans son étiquette
- Le routeur LSR change également la valeur de l'étiquette qu'il fait suivre.
- La valeur de l'étiquette n'est significative que pour deux équipements (LER-LSR, LSR-LSR, LSR-LER)
- Le traitement que doit effectuer un LER ou LSR est décrit dans une structure de données propre à chaque routeur MPLS appelée LIB (label Information base)

141

VPN MPLS

- Entête 32 bits
- Etiquette 20 bits
- Exp 3 bits fonctions expérimentales pour la Cos (Class of services)
- Bit S pile des étiquettes
- TTL 8 bits Time to Live



142

VPN MPLS

- Distributions des étiquettes MPLS
  - LSR commutation à base des étiquettes entre deux équipements voisins
  - Les LER et LSR doivent préalablement se mettre d'accord sur les traitements associés à chaque étiquette : utilisation d'un protocole de signalisation :
    - LSP Label Distribution Protocol
    - MP-BGP4 Multiprotocol-Border Gateway Protocol
    - OSPF Open Shortest Path First

143

VPN MPLS distribution des étiquettes

- Distribution non-centralisée : chaque équipement établit et annonce les informations de commutation en écoutant les annonces de routage
- Distribution contrôlée : un équipement du réseau MPLS est responsable de la distribution des informations de commutation
- La première approche a l'avantage de converger rapidement par rapport à la seconde. Par contre, elle présente l'inconvénient d'être plus difficile à utiliser l'ingénierie du réseau.

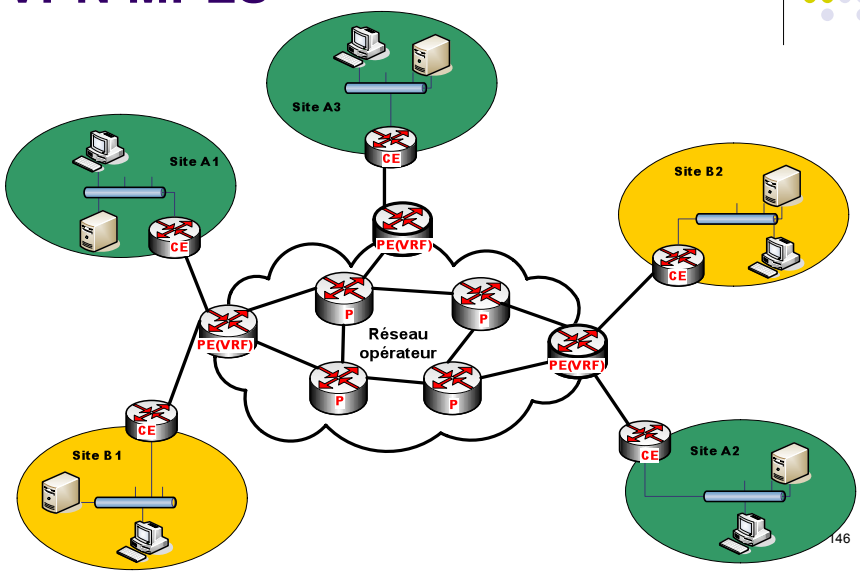
144

VPN MPLS

- Utilisation des étiquettes hiérarchiques
- RFC 2547 BGP MPLS VPN
  - http://www.faqs.org/rfcs/rfc2547.html
- Les équipements assurant la connectivité entre les PE (Provider edge) sont appelés P (provider router)
- Les CE sont sous la responsabilité des clients, les P et PE sont sous celle de l'opérateur
- Les PE maintiennent une table de commutation appelée VRF (VPN Routing and Forwarding table)
- A chaque VPN (donc à chaque client) correspond une table VRF spécifique au niveau du PE.
- Echange des info de routage VPN avec les autres PE en utilisant un protocole du type IBGP (Ingress BGP)
- Le PE transmet les données du trafic VPN au routeurs P en utilisant MPLS

145

VPN MPLS



146

VPN MPLS

- Trois types de flux transitent:
  - Les données que les clients échangent au sein du VPN
  - Les informations de contrôle relative aux VPN et échangées entre les PE
  - Les informations de contrôle relative aux chemins entre les PE(routage, étiquette)
- La sécurité des VPN MPLS se base sur la confiance dans le réseau de l'opérateur. Si des besoins forts de sécurité exigent le service de confidentialité, il est possible de combiner IPsec à MPLS

147

Mise en place IPSEC phase 1

- **Step 1** - Determine IKE phase one policy. Determine the IKE policies between IPSec peers based on the number and location of the peers . Some planning steps include the following:
  - Determine the key distribution method
  - Determine the authentication method
  - Identify IPSec peer IP addresses and host names
  - Determine ISAKMP policies for peers
- **Step 2** - Determine IKE phase two policy . Identify IPSec peer details such as IP addresses, IPSec transform sets, and IPSec modes , . Crypto maps will be used to gather all IPSec policy details together during the configuration phase .
- **Step 3** - Check the current configuration . Use the **show running-configuration**, **show isakmp [policy]**, and **show crypto map** commands. Other **show** commands can be used to check the current configuration of the router. This is covered later in this module.
- **Step 4** - Ensure that the network works without encryption . This step should not be avoided. Ensure that basic connectivity has been achieved between IPSec peers using the desired IP services before configuring IPSec. Use the **ping** command to check basic connectivity.
- **Step 5** - Ensure that the ACLs on perimeter devices are compatible with IPSec. Ensure that perimeter routers and the IPSec peer router interfaces permit IPSec traffic. Use the **show access-lists** command for this step

148

Étapes IPSEC

- Interesting traffic initiates the IPSec process. Traffic is deemed interesting when a packet triggers an access list that defines traffic to be protected.
- During IKE Phase One, IKE authenticates IPSec peers and negotiates IKE SAs, setting up a secure communications channel for negotiating IPSec SAs in phase two.
- During IKE Phase Two, IKE negotiates IPSec SA parameters and sets up matching IPSec SAs in the peers. These security parameters are used to protect data and messages exchanged between endpoints.
- During the data transfer phase, data is transferred between IPSec peers based on the IPSec parameters and keys stored in the SA database.
- During IPSec tunnel termination, IPSec SAs terminate through deletion or by timing out.

149

Mise en place IPSEC phase 2

- Enable IKE with the **crypto isakmp enable** command .
- Create IKE policies with the **crypto isakmp policy** commands - .
- First, configure the IKE identity to use the hostname or IP address . Configure pre-shared keys with the **crypto isakmp key** and associated commands .
- Verify the IKE configuration with the **show crypto isakmp policy** command .

150

Mise en place IPSEC phase 3

- Configure transform set suites with the **crypto IPSec transform-set** command.
- Configure global IPSec security association lifetimes with the c**rypto IPSec security-association lifetime** command.
- Configure crypto ACLs with the **access-list** command.
- Configure crypto maps with the **crypto map** command.
- Apply the crypto maps to the terminating/originating interface with the **interface** and **crypto map** commands.

151

Mise en place IPSEC phase 4

- Tester et vérifier IPSEC

152

VPN L2F

- L2F (Layer 2 Forwarding) VPN, CISCO 1998
- Le protocole L2F est employé pour établir un tunnel à travers une infrastructure publique (telle que l'Internet par téléphone) qui relie votre ISP à une passerelle à la maison. Ce tunnel crée un point virtuel pour diriger le trafic entre l'utilisateur et le réseau du client en entreprise.
- L2F permet la création d'un tunnel à la couche link (HDLC, async HDLC, ou SLIP frame) des protocoles de niveau plus élevé. En utilisant de tels tunnels, il est possible de faire transiger les communications, non pas de l'endroit du serveur initial d'appel téléphonique mais bien de l'endroit auquel le raccordement L2F est activé.

153

VPN L2F

- L2F permet l'encapsulation des paquets de la PPP /SLIP dans L2F. Les équipements de l'ISP et ceux à la maison exigent une connaissance commune du protocole d'encapsulation de sorte que les paquets de SLIP/PPP puissent être transmis et reçus à travers l'Internet avec succès.
- Les fonctions principales du Cisco L2F sont couvertes par L2TP, qui est le protocole standard d'IETF pour les tunnels.

154

VPN L2F

1	1	1	1	1	1	1	1	1	1	1	1	16	24	32bit
F	K	P	S	0	0	0	0	0	0	0	C	Version	Protocol	Sequence
Multiplex ID													Client ID	
Length													Offset	
Key														

- **Version :** The major version of the L2F software creating the packet.
- **Protocol :** The protocol field specifies the protocol carried within the L2F packet.
- **Sequence :** The sequence number is present if the S bit in the L2F header is set to 1.
- **Multiplex ID :** The packet multiplex ID identifies a particular connection within a tunnel.
- **Client ID :** The client ID (CLID) assists endpoints in demultiplexing tunnels.
- **Length :** The length is the size in octets of the entire packet, including the header, all the fields and the payload.
- **Offset :** This field specifies the number of bytes past the L2F header at which the payload data is expected to start. This field is present if the F bit in the L2F header is set to 1.
- **Key :** The key field is present if the K bit is set in the L2F header. This is part of the authentication process.
- **Checksum :** The checksum of the packet. The checksum field is present if the C bit in the L2F header is set to 1.

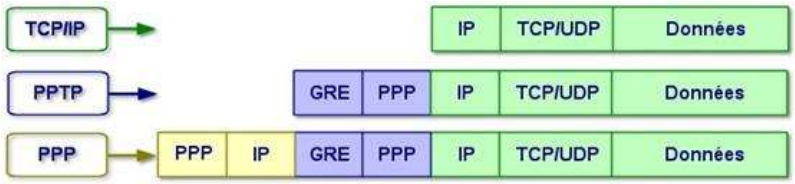
155

PPTP (Microsoft) (Point to Point Tunnel Protocol)

- Le principe du protocole PPTP est de créer des paquets sous le protocole PPP et de les encapsuler dans des datagrammes IP. PPTP crée ainsi un tunnel de niveau 3 défini par le protocole GRE (Generic Routing Encapsulation).
- Le tunnel PPTP se caractérise par une initialisation du client, une connexion de contrôle entre le client et le serveur ainsi que par la clôture du tunnel par le serveur. Lors de l'établissement de la connexion, le client effectue d'abord une connexion avec son fournisseur d'accès Internet.
- Cette première connexion établie une connexion de type PPP et permet de faire circuler des données sur Internet.
- Par la suite, une deuxième connexion dial-up est établie. Elle permet d'encapsuler les paquets PPP dans des datagrammes IP. C'est cette deuxième connexion qui forme le tunnel PPTP. Tout trafic client conçu pour Internet emprunte la connexion physique normale, alors que le trafic conçu pour le réseau privé distant, passe par la connexion virtuelle de PPTP.

156

PPTP (Microsoft)
(Point to Point Tunnel Protocol)



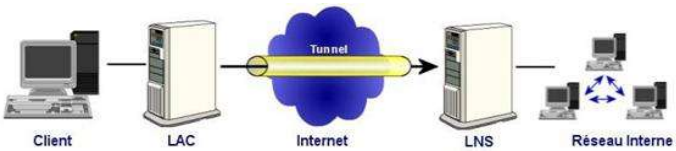
157

PPTP (Microsoft)
(Point to Point Tunnel Protocol)

- Plusieurs protocoles peuvent être associés à PPTP afin de sécuriser les données ou de les compresser. On retrouve évidement les protocoles développés par Microsoft et cités précédemment. Ainsi, pour le processus d'identification, il est possible d'utiliser les protocoles PAP (Password Authentification Protocol) ou MsChap.
- Pour l'encryptage des données, il est possible d'utiliser les fonctions de MPPE (Microsoft Point to Point Encryption). Enfin, une compression de bout en bout peut être réalisée par MPPC (Microsoft Point to Point Compression).

158

VPN L2TP



- L2tp repose sur deux concepts :
  - les concentrateurs d'accès L2tp (LAC : L2tp Access Concentrator)
  - les serveurs réseau L2tp (LNS : L2tp Network Server).
- L2tp n'intègre pas directement de protocole pour le chiffrement des données. C'est pourquoi L'IETF préconise l'utilisation conjointe d'Ipsec et L2tp.

159

VPN L2TP

- Ce protocole peut également être employé pour résoudre une problématique de vitesse de lien. Le PPP Multilink, est souvent employée pour agréger les liens RNIS B car il permet aux canaux composant le lien RNIS multilink d'être groupés à un serveur d'accès réseau (LNS).
- Puisque L2TP fait en sorte qu'une session PPP puisse apparaître à un endroit différent que le point physique auquel la session a été physiquement reçue, elle peut être employée pour faire en sorte que tous les canaux apparaissent à un LNS identique et ce, même si en réalité les connections physiques sont sur des concentrateurs différents.

160

VPN L2TP

- L2tp, définit par la Rfc 2661, est issu de la convergence des protocoles PPTP et L2F. Il est actuellement développé et évalué conjointement par Cisco Systems, Microsoft, Ascend, 3Com ainsi que d'autres acteurs clés du marché des réseaux.
- Il permet l'encapsulation des paquets ppp au niveau des couches 2 (Frame Relay et Atm) et 3 (ip). Lorsqu'il est configuré pour transporter les données sur IP, L2tp peut être utilisé pour faire du tunneling sur Internet.

161

VPN L2TP
Concentrateurs d'accès LAC

- LAC : L2tp Access Concentrator
- Les périphériques LAC fournissent un support physique aux connexions L2tp. Le trafic étant alors transféré sur les serveurs réseau L2tp. Ces serveurs peuvent s'intégrer à la structure d'un réseau commuté Rtc ou alors à un système d'extrémité ppp prenant en charge le protocole L2tp. Ils assurent le fractionnement en canaux de tous les protocoles basés sur ppp.
- Le LAC est l'émetteur des appels entrants et le destinataire des appels sortants.

162

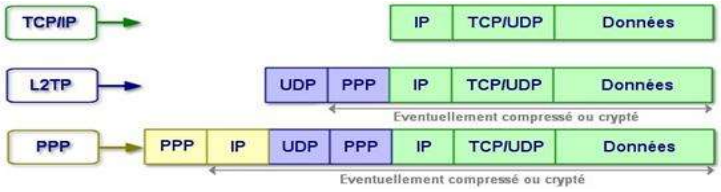
VPN L2TP
Serveur réseau Lns

- **LNS : L2tp Network Server**
- Les serveurs réseau L2tp ou LNS peuvent fonctionner sur toute plate-forme prenant en charge la terminaison Ppp.
- Le Lns gère le protocole L2tp côté serveur. Le protocole L2tp n'utilise qu'un seul support, sur lequel arrivent les canaux L2tp. C'est pourquoi, les serveurs réseau LNS, ne peuvent avoir qu'une seule interface de réseau local (Lan) ou étendu (Wan). Ils sont cependant capables de terminer les appels en provenance de n'importe quelle interface ppp du concentrateur d'accès LAC : async., Rnis, ppp sur Atm ou ppp sur relais de trame.
- Le LNS est l'émetteur des appels sortants et le destinataire des appels entrants.
- C'est le LNS qui sera responsable de l'authentification du tunnel.

163

VPN L2TP

- On distingue principalement 2 composantes dans L2tp:
  - Les paquets d'information, encapsulés dans des paquets Ppp pour les sessions utilisateurs qui servent pour le transport de L2tp.
  - Le protocole de signalisation, qui utilise le contrôle de l'information L2tp est encapsulé dans des paquets udp/ip



164

OpenVPN

- OpenVPN est un outil de création de VPN (Virtual Private Network, ou réseau virtuel privé) implémentant la sécurité au niveau des couches du modèle OSI 2 et 3 en utilisant le standard SSL/TLS.
- Il est facile à utiliser, robuste et grandement configurable. Il peut être utilisé pour relier de manière sécurisée deux ou plus réseaux privés, en utilisant un tunnel chiffré à travers Internet.
- OpenVPN fonctionne sous les Unix (Linux, BSD, etc) et sous les Windows et par surcroît est GPL.

165

OpenVPN

Avec OpenVPN on peut :

- encapsuler dans un tunnel n'importe quel sous-réseau IP ou adaptateur Ethernet virtuel, dans un unique port TCP ou UDP;
- créer une infrastructure de tunnels entre n'importe quel système d'exploitation supporté par OpenVPN. Sont concernés Linux, Solaris, OpenBSD, FreeBSD, NetBSD, Mac OS X, et Windows 2000/XP;
- utiliser toutes les fonctionnalités de chiffrement, d'authentification et de certification de la librairie OpenSSL afin de protéger le trafic de votre réseau privé lorsqu'il transite par Internet;
- utiliser n'importe quel algorithme de chiffrement, taille de clef, ou empreinte HMAC (pour l'authentification des datagrammes) supporté par la bibliothèque OpenSSL,

166

OpenVPN

- choisir entre un chiffrement conventionnel basé sur une clef statique, ou un chiffrement par clef publique en se basant sur les certificats,
- utiliser les clefs statiques, pré-partagée (PSK), ou un échange de clef dynamiques basé sur TLS,
- utiliser une compression des flux adaptée en temps-réel, la mise en forme de trafic pour gérer l'utilisation de la bande-passante du lien,
- encapsuler les réseaux dont les extrémités publiques sont dynamiques comme on peut le rencontrer avec DHCP ou avec des clients connectés par modem,
- encapsuler le trafic de réseaux grâce aux firewalls effectuant du suivi de sessions, sans nécessairement utiliser des règles de filtrage particulières,
- encapsuler le trafic de réseaux avec du NAT (translation d'adresse),
- créer des ponts ethernet sécurisés utilisant les périphériques tap.

167

OpenVPN

Qu'est-ce qui distingue OpenVPN des autres VPN ?

- Les principaux points forts de OpenVPN est de proposer la portabilité inter-plateforme pour la plupart des environnements informatiques connus, une excellente stabilité, une installation relativement aisée qui dans la plupart des cas ne requiert pas un module noyau spécifique, puis le support des adresses IP dynamiques et le NAT.
- OpenVPN semble être le seul produit VPN open source à supporter entièrement l'ICP (Infrastructure à Clef Publique) OpenSSL pour la session d'authentification, le protocole TLS pour l'échange de clef, l'interface EVP (indépendante du chiffrement utilisé) fournie par OpenSSL pour chiffrer les données encapsulées, l'algorithme HMAC pour authentifier les données encapsulées, et pour multiplexer tout ceci au travers d'un unique port UDP.

168

OpenVPN

- OpenVPN utilise un modèle de sécurité avec une robustesse équivalente à ce que l'on retrouve dans l'industrie. Ce modèle est conçu pour protéger à la fois contre les attaques passives et contre les attaques actives. Le modèle de sécurité de OpenVPN est similaire à celui de IPSec mais avec un encombrement bien moindre et sans nécessiter une modification du noyau ou de la pile IP.
- OpenVPN supporte :
  - le choix, pour le chiffrement, de n'importe quel type de chiffrement de OpenSSL, et la taille de la clef,
  - le choix des modes de chiffrement CBC, CFB, et OFB,
  - HMAC pour l'authentification des datagrammes, l'indication explicite d'un IV (Initialization Vector ou Vecteur d'Initialisation) pour masquer le modèle dans le texte clair,
  - protection anti-*replay* utilisant un horodateur/numéro-de-séquence unique pour identifier le datagramme, et,
  - le protocole TLS pour authentifier les sessions en se basant sur des certificats.
  - En outre, n'importe laquelle de ces options peut être désactivée pour améliorer les performances. Le coût de cette amélioration de performances est la baisse du niveau de sécurité.

169

OpenVPN

- OpenVPN a été construit pour être portable. OpenVPN tourne sous Linux, Solaris, OpenBSD, FreeBSD, NetBSD, Mac OS X, et Windows 2000/ XP. Étant donné que OpenVPN est écrit pour fonctionner comme un démon en espace utilisateur plutôt que comme un module du noyau ou comme une modification complexe de la couche IP, les efforts pour porter le logiciel ont été considérablement simplifiés.
- OpenVPN est facile à utiliser. Généralement, un tunnel peut être créé et configuré avec une seule commande (et sans nécessiter de fichiers de configuration).

170

OpenVPN

- OpenVPN a été rigoureusement conçu et testé pour être robuste sur des réseaux incertains. Le principal effort de conception d'OpenVPN est de faire en sorte qu'il soit aussi réactif, à la fois lors d'opérations normales et de reprise sur erreur, que la couche IP sous-jacente qu'il encapsule. Cela signifie que si la couche IP tombe pendant 5 minutes, quand elle remonte, le trafic du tunnel sera immédiatement repris même si la panne a interféré avec l'échange de clef dynamiques qui était réalisé pendant ce temps.
- OpenVPN a été conçu avec une conception très modulaire. Tout ce qui concerne le chiffrement est manipulé par la bibliothèque OpenSSL, et toutes les fonctionnalités d'encapsulation IP sont fournies par les pilotes du réseau virtuel TUN/TAP.

171

OpenVPN

- OpenVPN supporte le multithreading basé sur « pthread » pour optimiser la latence de l'échange de clef dynamiques SSL/TLS. Cela vous permet l'utilisation de longues clef RSA (tel que 2048 bits ou plus) et de les renégocier fréquemment avec peu ou pas de baisse de performance.
- OpenVPN est flexible. Vous pouvez créer autant de tunnels vers ou depuis la même machine, vous pouvez appliquer de la mise en forme du trafic du tunnel pour limiter l'usage de la bande-passante, et vous pouvez choisir graduellement entre la sécurité et l'efficacité en contrôlant exactement combien de rajouts en relation avec la sécurité sont ajoutés à chaque datagramme.

172

OpenVPN

Pourquoi choisir TLS avec OpenVPN ?

- TLS est la dernière évolution de la famille SSL, protocoles qui ont été développés au départ par Netscape pour leur premier navigateur web sécurisé. TLS et ses prédécesseurs SSL ont vu leur utilisation se répandre sur le web pendant plusieurs années et ont été analysés avec intensité pour détecter leurs faiblesses. Alternativement, cette analyse a mené à un renforcement conséquent du protocole tel qu'on le connait aujourd'hui, SSL/TLS est considéré pour être un des plus résistant et des plus matures protocoles de sécurité disponible

173

OpenVPN

- http://openvpn.net/
- http://openvpn.se/ -> Interface graphique pour windows
- http://www.itsatechworld.com/2006/01/29/how-to-configure-openvpn/

174

Ssh

- Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé.
- SSH (rfc 4251) est composé de :
  - La couche transport (rfc 4253),
  - La couche authentification utilisateur (rfc 4252)
  - La couche connection (rfc 4254)

175

Ssh sous Linux

- Implémentation libre Openssh
- Possibilité de faire un tunnel chiffré sous tcp (port forwarding)

  ☹Attention aux tunnels tcp over tcp

  $>ssh –L <port local>:<machine cible>:<port cible> <machine cible>

  ssh -N -L 3128:proxy.univ-pau.fr:3128 proxyuser@proxy.univ-pau.fr

176