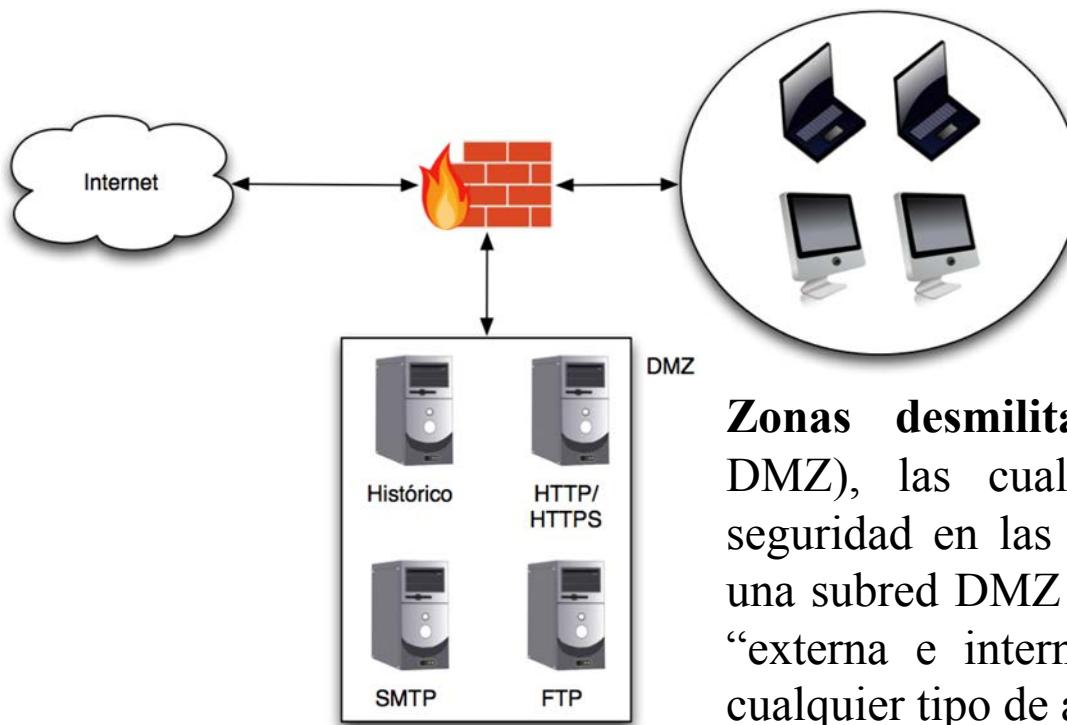


FIREWALLS EN REDES

¿Qué es un cortafuego?

Firewalls

- Un **cortafuego (firewall)** es un sistema (software o hardware) que establece un conjunto de políticas de control
- El espacio protegido, denominado **perímetro de seguridad**, suele ser propiedad de la misma organización, y la protección se realiza generalmente contra una red externa (ej. el Internet) no confiable, llamada **zona de riesgo**



Zonas desmilitarizadas (De-Militarized Zones -- DMZ), las cuales añaden un nivel específico de seguridad en las arquitecturas de cortafuegos situando una subred DMZ (basada en servidores) entre las redes “externa e interna”, de forma que aísla y/o protege cualquier tipo de acceso a los hosts del sistema

IPTables

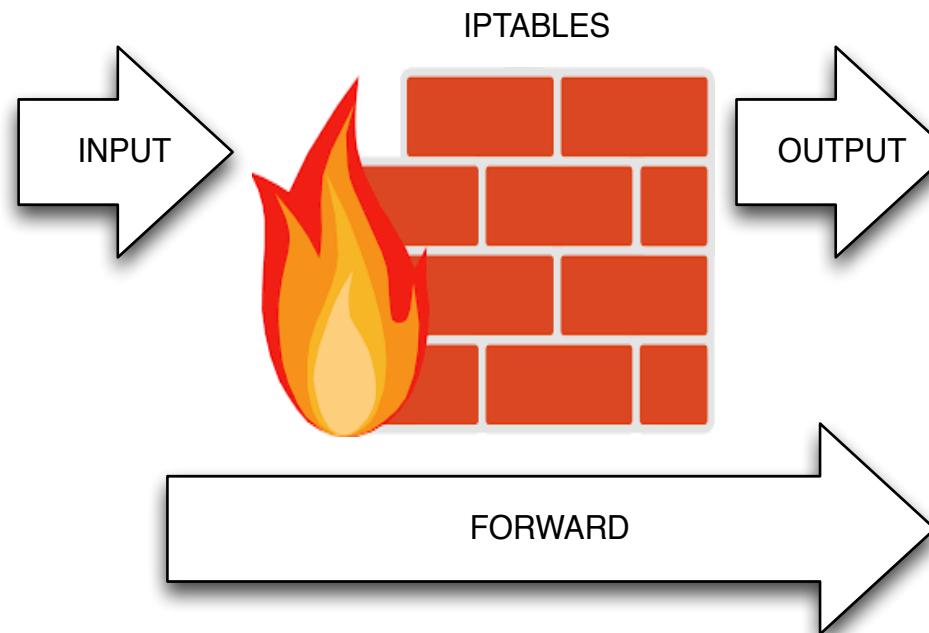
- IPTables es un sistema de firewall basado en reglas, desarrollado para el kernel de Linux cuyas reglas se ejecutan a través del comando **IPTables**
 - Tipos de reglas:
 - añadir, borrar o crear reglas para tráfico entrante y/o saliente
 - Objetivo:
 - Habilitar el acceso a puertos específicos y a determinadas IPs
 - permitiendo no sólo el acceso a tráfico TCP sino UDP, ICMP y otros
 - Denegar el acceso desde redes externas (o interna) a puertos específicos
 - Enmascarar tráfico de red local hacia redes públicas o externas

IPTABLES: sintaxis y políticas

- Iptables ilustra la siguiente sintaxis de comandos:

```
|[root@localhost alumno]# #iptables -A <chain> -j <target>|
```

- <chain>: define una cadena de reglas que pueden ser del tipo INPUT, OUTPUT y FORWARD



IPTABLES: sintaxis y políticas

- Iptables ilustra la siguiente sintaxis de comandos:

```
[[root@localhost alumno]# #iptables -A <chain> -j <target>]
```

- <chain>: define una cadena de reglas que pueden ser del tipo INPUT, OUTPUT y FORWARD
- -A <chain>: agrega la regla a la cadena
 - -D <chain>: elimina la regla
 - -I <chain>: inserta una nueva regla en una posición determinada
- -j <target>: especifica el fin de la regla; es decir, qué hacer si un paquete coincide con la regla, como, por ejemplo: ACCEPT y DROP

- **Eliminar cualquier tráfico telnet entrante:**
iptables -I INPUT -p tcp --dport 23 -j DROP
- **Eliminar cualquier tráfico web saliente:**
iptables -I OUTPUT -p tcp --dport 80 -j DROP
- **Eliminar cualquier tráfico saliente a la IP 192.168.0.1**
iptables -I OUTPUT -p tcp --dest 192.168.0.1 -j DROP
- **Permitir cualquier tráfico web entrante:**
iptables -I INPUT -p tcp --dport 80 -j ACCEPT
- **Permitir tráfico entrante del puerto 25 (del servidor SMTP):**
iptables -A INPUT -s 0.0.0.0/0 -p tcp --dport 25 -j ACCEPT
- **Permitir tráfico pop3 entrante:**
iptables -A INPUT -s 0.0.0.0/0 -p tcp --dport 110 -j ACCEPT

- **Permitir tráfico HTTPS (443) entrante desde la IP 11.3.2.1**
iptables -I INPUT -s 11.3.2.1 -p tcp --dport 443 -j ACCEPT
- **Denegar el tráfico saliente a la red 192.2.4.0-192.2.4.255:**
iptables -I OUTPUT -d 192.2.4.6.0/24 -j DROP
- **Bloquear cualquier tráfico saliente de un particular dominio o host:**
 - 1) host -t a elpais.es ➔ elpais.es tiene IP:78.120.152.203
 - 2) iptables -A OUTPUT -d 78.120.152.203 -j DROP

IPTABLES: flush y políticas por defecto

- Política establecida por defecto:
 - Para cada cadena <chain> se define una política inicial que puede ser ACCEPT o DROP, y para ello se usa la opción –P

```
### FLUSH de reglas
```

```
iptables -F  
iptables -X  
iptables -Z  
iptables -t nat -F
```

```
### Políticas por defecto
```

```
iptables -P INPUT ACCEPT  
iptables -P OUTPUT ACCEPT  
iptables -P FORWARD ACCEPT  
iptables -t nat -P PREROUTING ACCEPT  
iptables -t nat -P POSTROUTING ACCEPT
```

IPTABLES: flush y políticas por defecto

FLUSH de reglas

```
iptables -F  
iptables -X  
iptables -Z  
iptables -t nat -F
```

Políticas por defecto

```
iptables -P INPUT ACCEPT  
iptables -P OUTPUT ACCEPT  
iptables -P FORWARD ACCEPT  
iptables -t nat -P PREROUTING ACCEPT  
iptables -t nat -P POSTROUTING ACCEPT
```



```
[alumno@router ~]$ su  
Contraseña:  
[root@router alumno]# iptables -nvL  
Chain INPUT (policy ACCEPT 20845 packets, 27M bytes)  
pkts bytes target prot opt in out source destination  
  
Chain FORWARD (policy ACCEPT 84 packets, 5676 bytes)  
pkts bytes target prot opt in out source destination  
  
Chain OUTPUT (policy ACCEPT 14649 packets, 1370K bytes)  
pkts bytes target prot opt in out source destination
```

```
[root@router alumno]# iptables -t nat -nvL  
Chain PREROUTING (policy ACCEPT 3 packets, 718 bytes)  
pkts bytes target prot opt in out source destination  
  
Chain INPUT (policy ACCEPT 3 packets, 718 bytes)  
pkts bytes target prot opt in out source destination  
  
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)  
pkts bytes target prot opt in out source destination  
  
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)  
pkts bytes target prot opt in out source destination
```

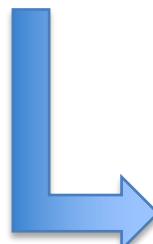
IPTABLES: INPUT y OUTPUT

- **Ejemplo 1:** no permitir producir señales de entrada y salida al firewall, y para todas las redes (incluyendo al Router)

```
[root@router alumno]# iptables -P INPUT DROP
[root@router alumno]# iptables -P OUTPUT DROP
[root@router alumno]# ping 192.168.0.15
PING 192.168.0.15 (192.168.0.15) 56(84) bytes of data.
^C
--- 192.168.0.15 ping statistics ---
0 packets transmitted, 0 received

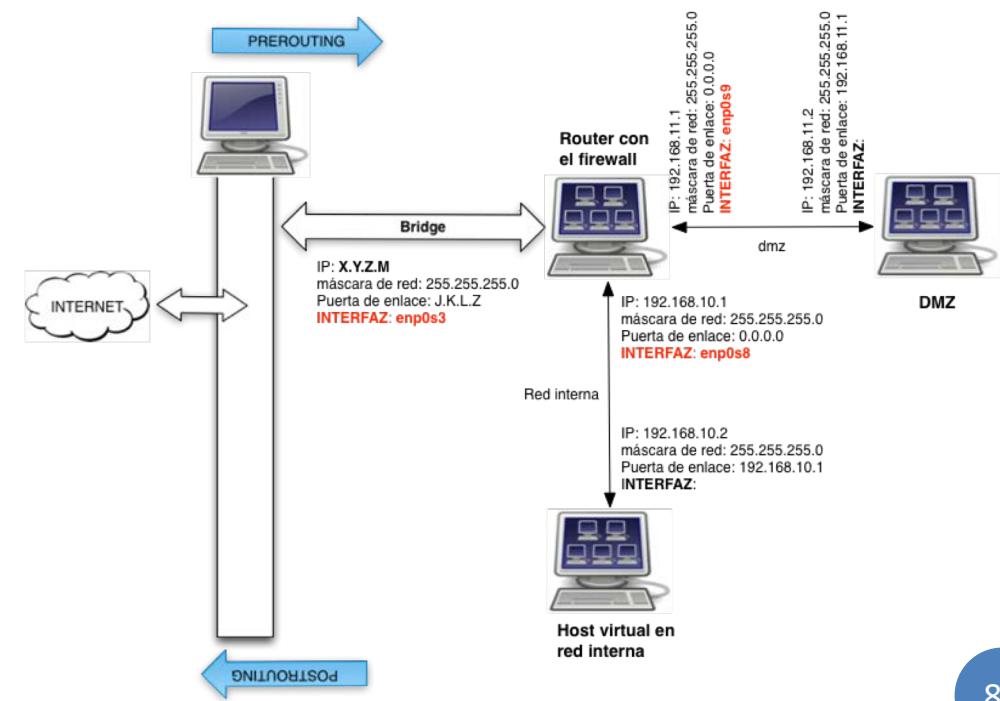
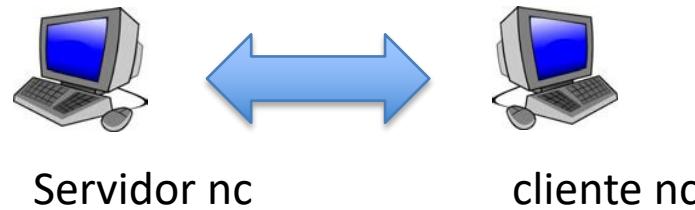
[root@router alumno]# ping 192.168.11.1
PING 192.168.11.1 (192.168.11.1) 56(84) bytes of data.
^C

--- 192.168.11.1 ping statistics ---
0 packets transmitted, 0 received
```



```
[root@router sysctl.d]# iptables -nL
Chain INPUT (policy DROP)
target      prot opt source          destination
Chain FORWARD (policy ACCEPT)
target      prot opt source          destination
Chain OUTPUT (policy DROP)
target      prot opt source          destination
```

- **Ejemplo 2:** crear un cliente-servidor **netcat** en las máquinas instaladas en la red DMZ y la red interna para que se comuniquen, sólo y únicamente, por el puerto 55555



- **Ejemplo 2:** crear un cliente-servidor netcat en las máquinas instaladas en la red DMZ y la red interna para que se comuniquen, sólo y únicamente, por el puerto 55555

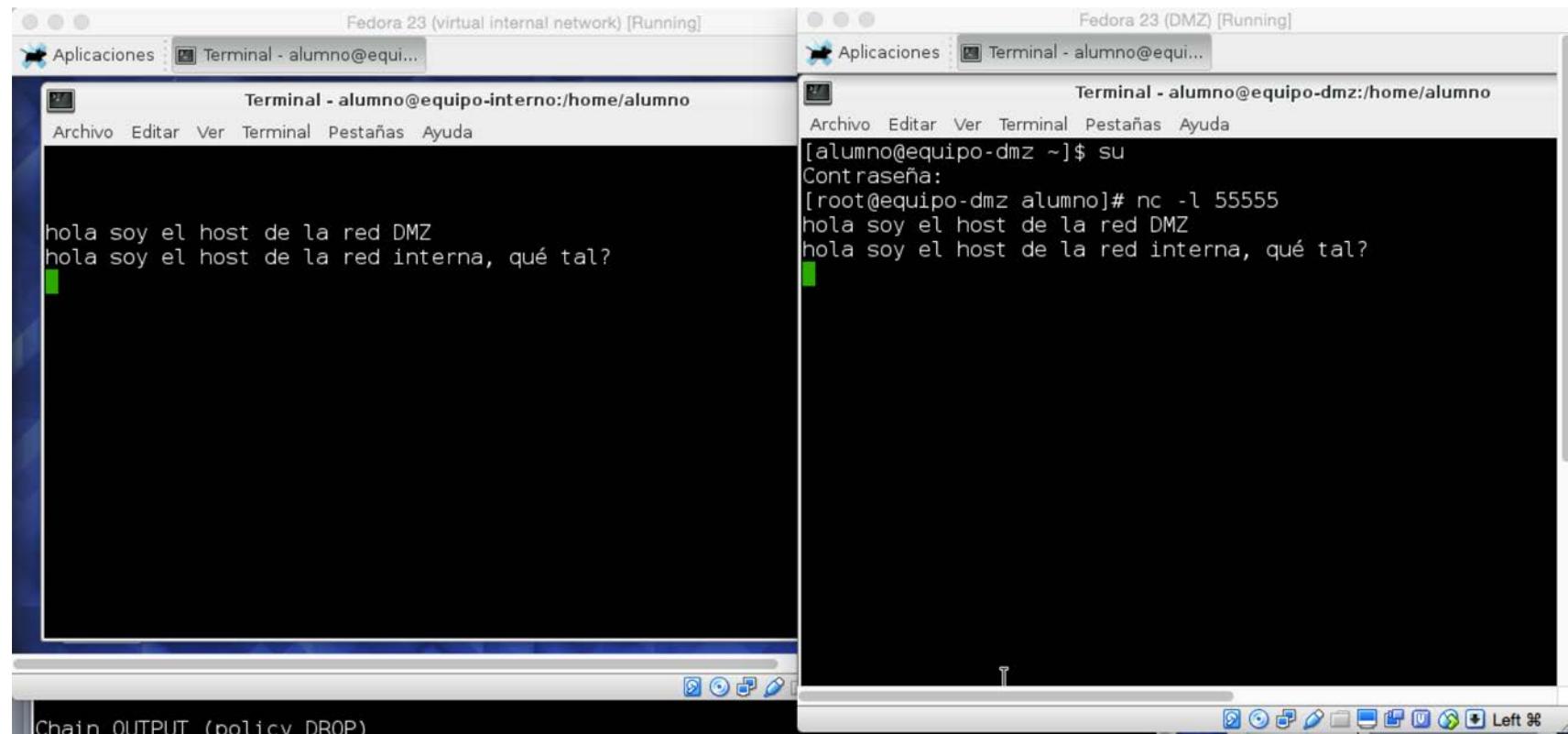
```
### ASEGURAR EL FORWARDING
sysctl -w net.ipv4.ip_forward=1
echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/ip_forwarding.conf

### FLUSH de reglas
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

### Políticas por defecto
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

## PROPORCIONAR LA COMUNICACIÓN ENTRE LAN-LAN PERO SOLO POR EL PUERTO 55555
iptables -A FORWARD -i enp0s9 -o enp0s8 -p tcp --dport 55555 -j ACCEPT
iptables -A FORWARD -i enp0s8 -o enp0s9 -p tcp --sport 55555 -j ACCEPT
```

IPTABLES: Netcat



Cliente nc – Router

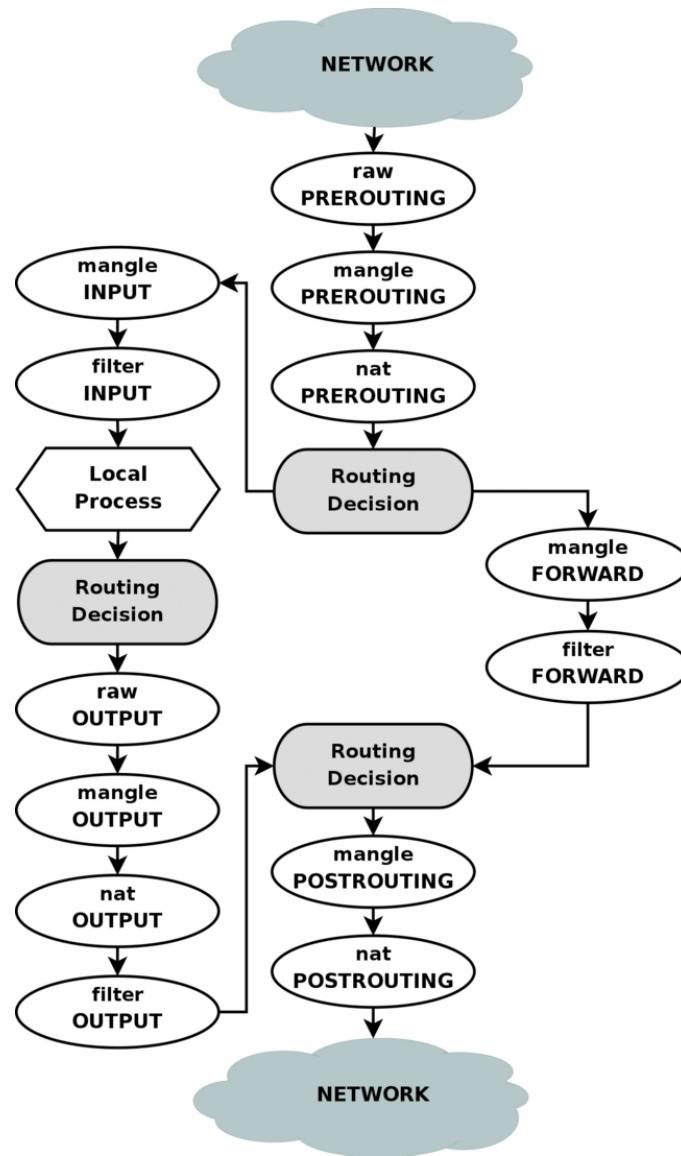
Servidor nc – Red DMZ

IPTABLES: PREROUTING Y POSTROUTING

- Aunque FORWARD permite el enrutamiento, la comunicación hacia o desde redes públicas requiere:
 - **POSTROUTING:** INTERIOR → EXTERIOR
 - Realizar un proceso de enmascaramiento para poder trabajar con el NAT
 - El orden de las reglas importa: primero los FORWARD y después el POSTROUTING
- **PREROUTING:** EXTERIOR → INTERIOR
 - El orden de las reglas importa: primero el PREROUTING y después los FORWARD

```
[root@router alumno]# iptables -t nat -A PREROUTING -i enp0s3 -d X.Y.Z.M -p tcp --dport 443 -j DNAT --to 192.168.11.2:443
```

IPTABLES: sintaxis y políticas



Ej. Conexión de una red local a Internet

```
### ASEGURAR EL FORWARDING
sysctl -w net.ipv4.ip_forward=1
echo "net.ipv4.ip_forward = 1" > /etc/sysctl.d/ip_forwarding.conf

### FLUSH de reglas
iptables -F
iptables -X
iptables -Z
iptables -t nat -F

### Políticas por defecto
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

### Restricción para la red interna y no para la red DMZ con iterfaz enps9
iptables -A FORWARD -i enp0s9 -o enp0s3 -j ACCEPT
iptables -A FORWARD -i enp0s3 -o enp0s9 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
#iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
#iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE

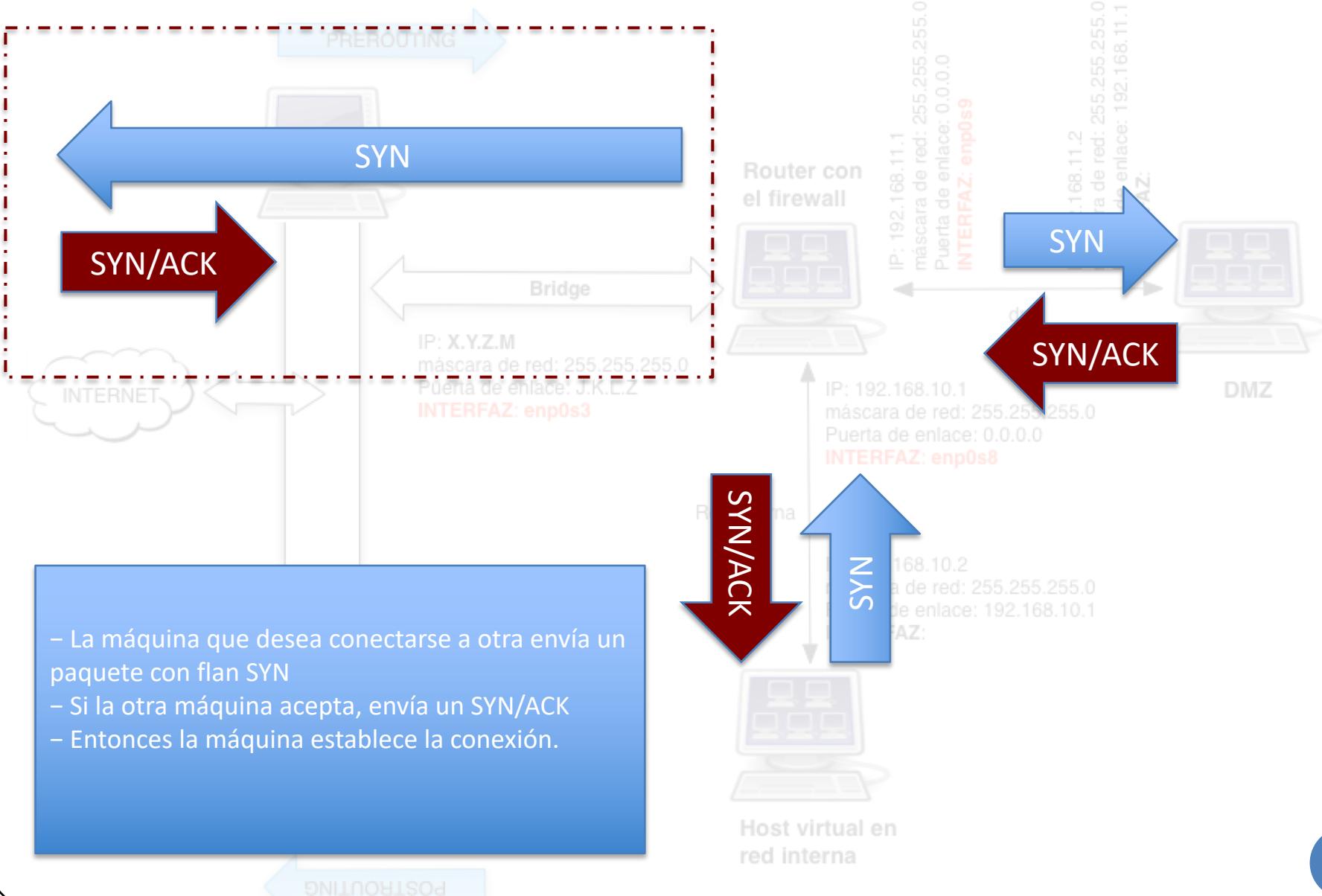
iptables -nvL
```

```
[root@router alumno]# chmod +x script-http.sh
[root@router alumno]# ./script-http.sh
Chain INPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target prot opt in     out      source          destination

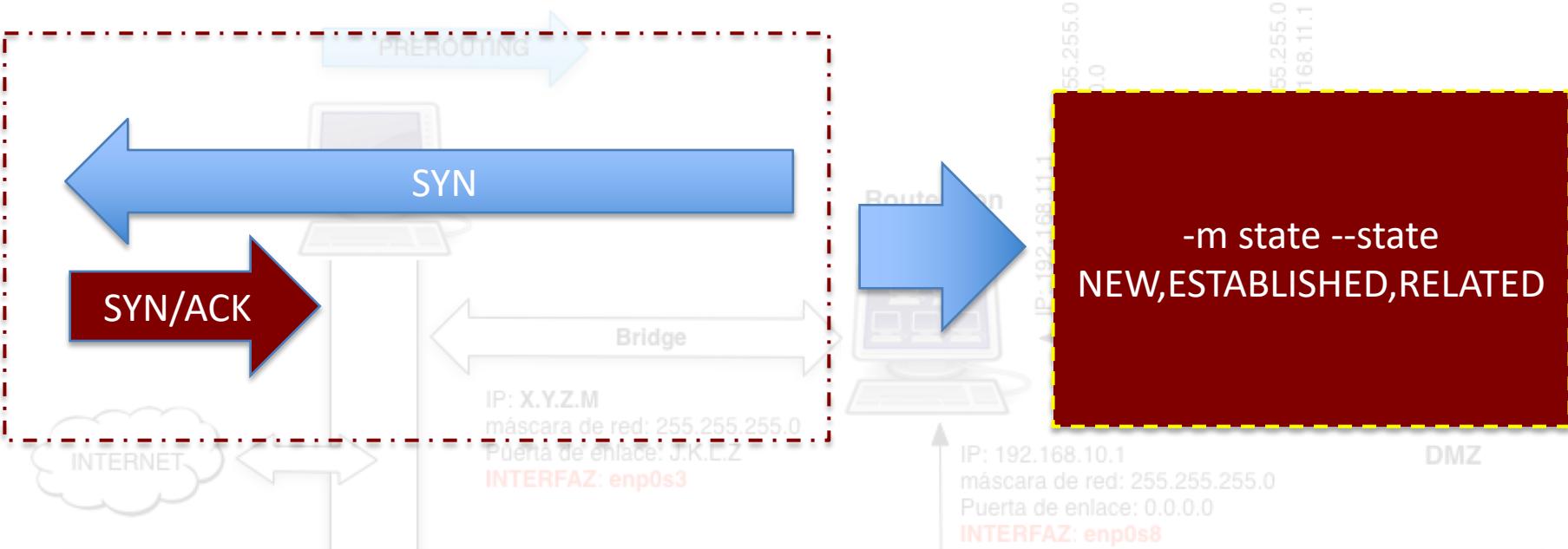
Chain FORWARD (policy DROP 0 packets, 0 bytes)
pkts bytes target prot opt in     out      source          destination
      0     0 ACCEPT   all   --  enp0s9  enp0s3  0.0.0.0/0    0.0.0.0/0
      0     0 ACCEPT   all   --  enp0s3  enp0s9  0.0.0.0/0    0.0.0.0/0
                                         state NEW,RELATED,ESTABLISHED

Chain OUTPUT (policy DROP 0 packets, 0 bytes)
pkts bytes target prot opt in     out      source          destination
```

Three-way-handshake en TCP/IP y problema a controlar



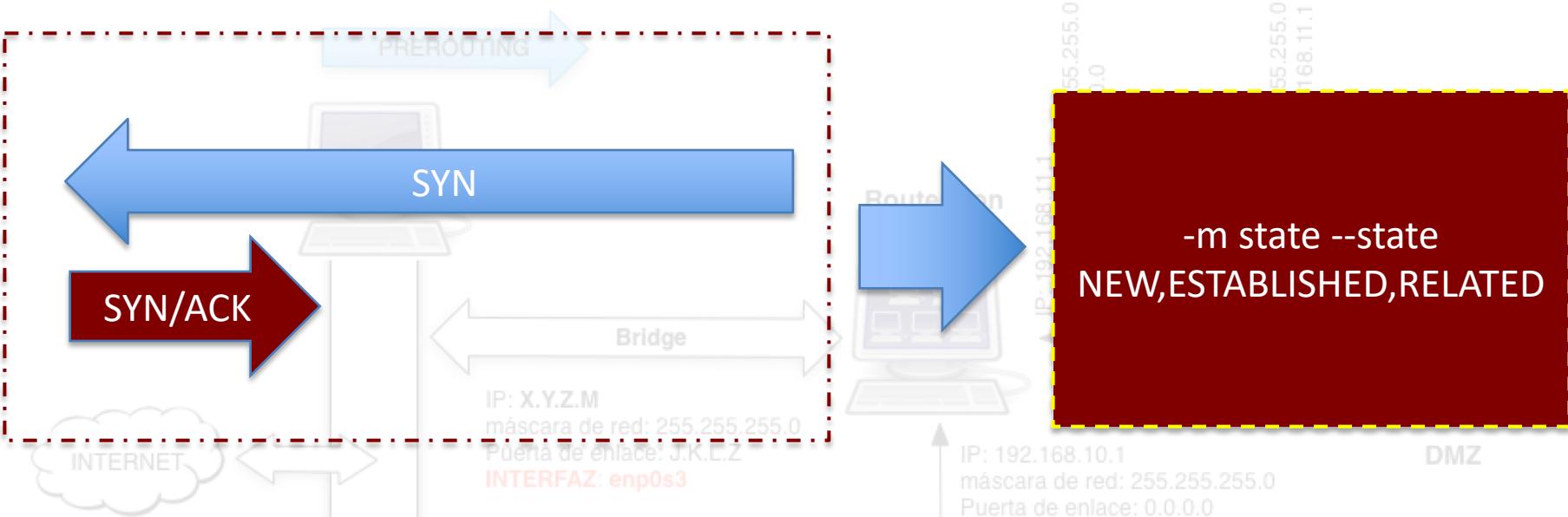
Three-way-handshake en TCP/IP y problema a controlar



- NEW: el paquete ha establecido una nueva conexión y se requiere comunicación bidireccional (ej. SYN -- SYN/ACK)
- ESTABLISHED: el paquete está asociado con una conexión ya establecida pero se requiere que haya paquetes en ambas direcciones (ej. SYN -- SYN/ACK)
- RELATED: el paquete está comenzando una nueva conexión pero está asociada a una conexión ya existente, como puede ser una comunicación FTP

Host virtual en
red interna

Three-way-handshake en TCP/IP y problema a controlar

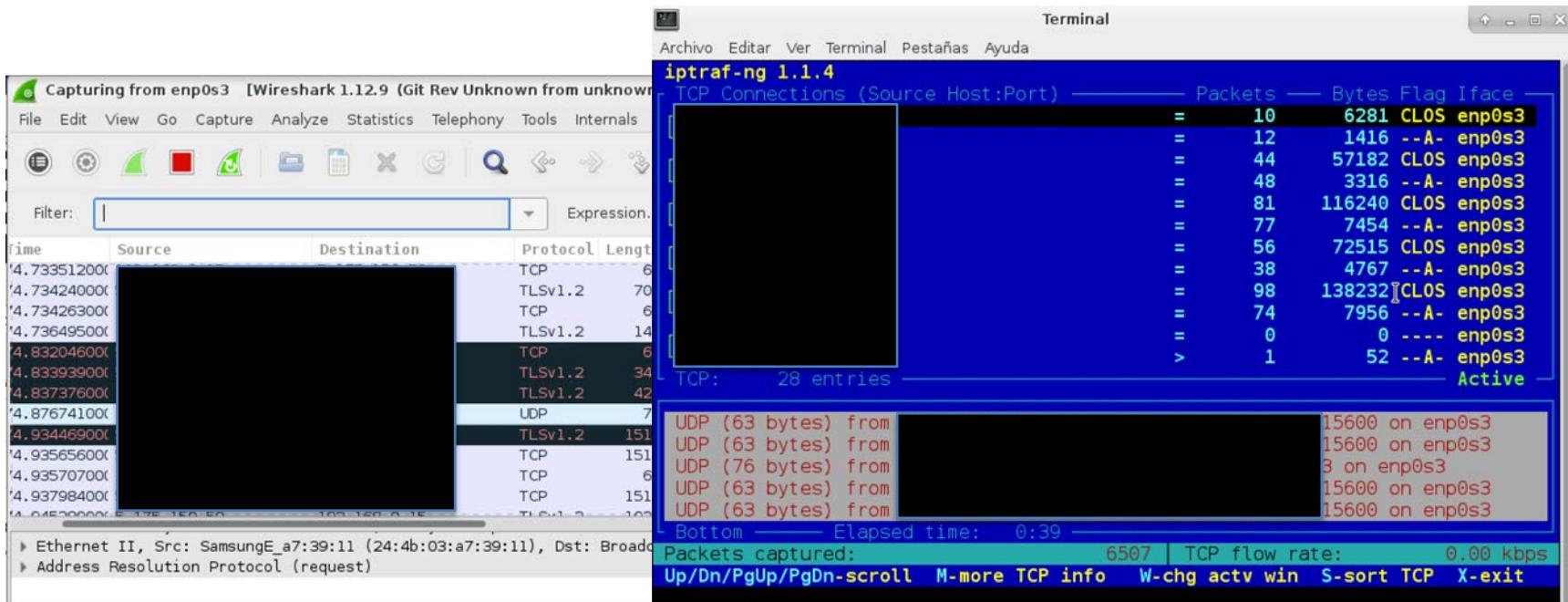


```
### Políticas por defecto
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT

### Permitiendo comunicación con la red externa
iptables -A FORWARD -i enp0s9 -o enp0s3 -j ACCEPT
iptables -A FORWARD -i enp0s3 -o enp0s9 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -i enp0s8 -o enp0s3 -j ACCEPT
iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE

iptables -nvL
```

Herramientas de monitorización



```
0000 ff ff ff ff ff ff [root@router pub]# tcpdump -D
0010 08 00 06 04 00 01 1.enp0s3 [Up, Running]
0020 00 00 00 00 00 00 2.enp0s8 [Up, Running]
0030 00 00 00 00 00 00 3.enp0s9 [Up, Running]
4.any (Pseudo-device that captures on all interfaces) [Up, Running]
5.lo [Up, Running, Loopback]
6.bluetooth-monitor (Bluetooth Linux Monitor)
7.usbmon1 (USB bus number 1)
[root@router pub]# tcpdump -i 3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s9, link-type EN10MB (Ethernet), capture size 262144 bytes
18:39:45.534659 IP [REDACTED] 934 > mad06s09-in-f3.1e100.net.https: Flags
., seq 1367477101:1367477132, ack 1352877774, win 507, options [nop,nop,TS va
38372341 ecr 32599377161 length 31
18:39:45.534906 IP [REDACTED] 34 > mad06s09-in-f3.1e100.net.https: Flags
., seq 31, ack 1, win 507, options [nop,nop,TS val 38372341 ecr 3259937716],
length 0
18:39:45.534938 IP [REDACTED] > mad06s09-in-f3.1e100.net.https: Flags
```

SEGURIDAD EN LA CAPA DE ACCESO A RED: EL CASO DE LAS REDES INALÁMBRICAS

Consideraciones generales

- Existe una amplia variedad de tecnologías y tipos de redes inalámbricas, entre las que destacan Wi-Fi, Bluetooth, WiMAX, Zigbee, etc.
- Los requisitos de seguridad de estas redes son los mismos que en el caso de las redes cableadas
- Sin embargo, hay algunas amenazas de seguridad que aumentan cuando se consideran las redes inalámbricas
 - y además, hay otras amenazas que son propias de estos entornos
- La fuente de riesgo más significativa en las redes inalámbricas es el medio de comunicación subyacente
 - pero también hay riesgos de seguridad en los propios protocolos inalámbricos cuando no se diseñan apropiadamente

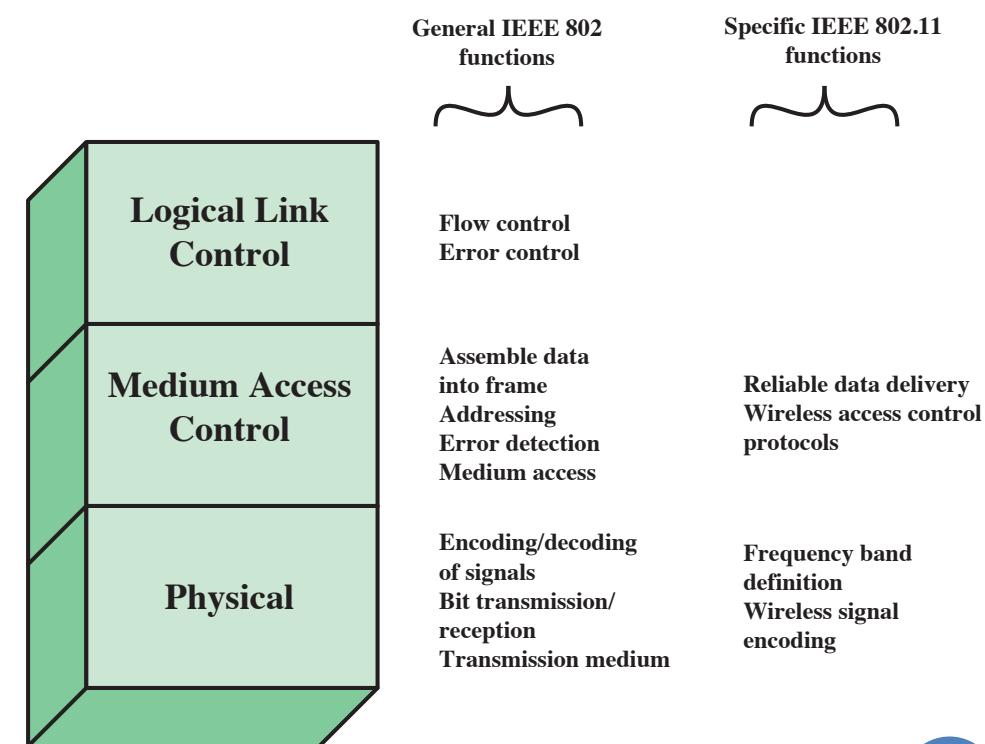
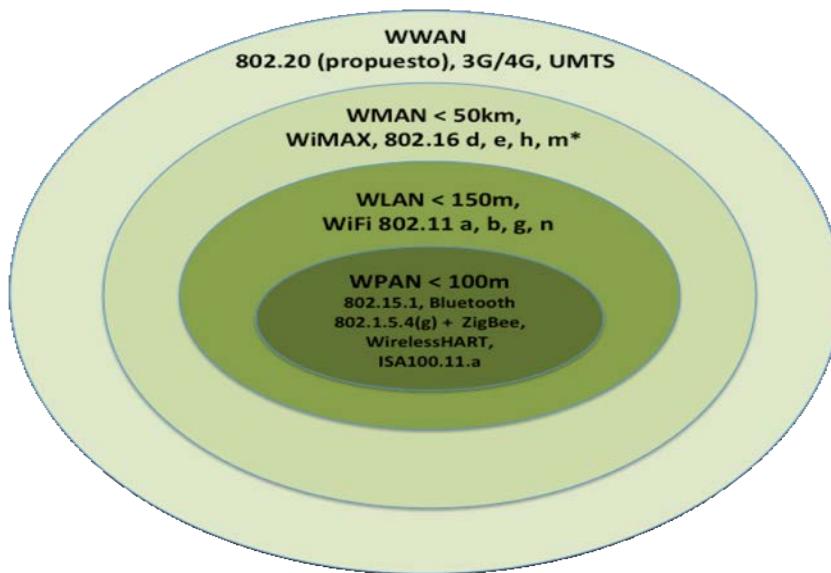
- A grandes rasgos, el entorno inalámbrico tiene tres puntos de ataque: **cliente** (o estación), **punto de acceso (AP)** y **medio de transmisión**
- Las posibles amenazas generales son:
 - Asociación accidental
 - Asociación intencionada
 - Robo de identidad (o sea, de la dirección MAC del dispositivo)
 - Ataque desde el intermedio (Man-in-the-middle)
 - Denegación de servicio
 - Inyección en la red

- En cuanto a **medidas de seguridad**, éstas se pueden clasificar según esas medidas estén relacionadas con:
 - la transmisión inalámbrica
 - el puntos de acceso (AP)
 - los propios elementos de la propia red, o sea, routers y equipos
- Medidas de seguridad relacionadas con las **transmisiones inalámbricas**:
 - las principales amenazas son la copia de mensajes, la alteración o inserción de mensajes, así como la interrupción de los mismos
 - las contramedidas son:
 - técnicas de ocultación de la señal
 - cifrado para el caso de la copias de mensajes
 - cifrado y autenticación para los casos de alteración o inserción
 - métodos contra DoS en el caso de interrupción

- Medidas de seguridad relacionadas con el **AP**:
 - la mayor amenaza que involucra al AP es el acceso no autorizado a la red
 - la forma de evitarlo es usando mecanismos de autenticación para los dispositivos que se quieren conectar a la red
- Medidas de seguridad relacionadas con **los elementos de la red**:
 - cifrado: normalmente integrado en los routers inalámbricos para el tráfico entre routers
 - deshabilitar el broadcast de identificación: sólo los dispositivos autorizados podrán conocer la identidad de los routers
 - dejar que sólo equipos específicos se conecten a la red: sólo direcciones cuyas direcciones MAC sean conocidas
 - cambiar el identificador por defecto que el router trae de fábrica
 - cambiar el password preestablecido para la administración del route

Redes inalámbricas IEEE 802.11

- IEEE 802 es un comité que ha desarrollado estándares para diferentes tipos de redes LAN
- IEEE 802.11 es un capítulo de ese comité, y su objetivo es el desarrollo de un protocolo y las especificaciones de transmisión para LANs inalámbricas



- Diferencias entre una red cableada y una red inalámbrica:

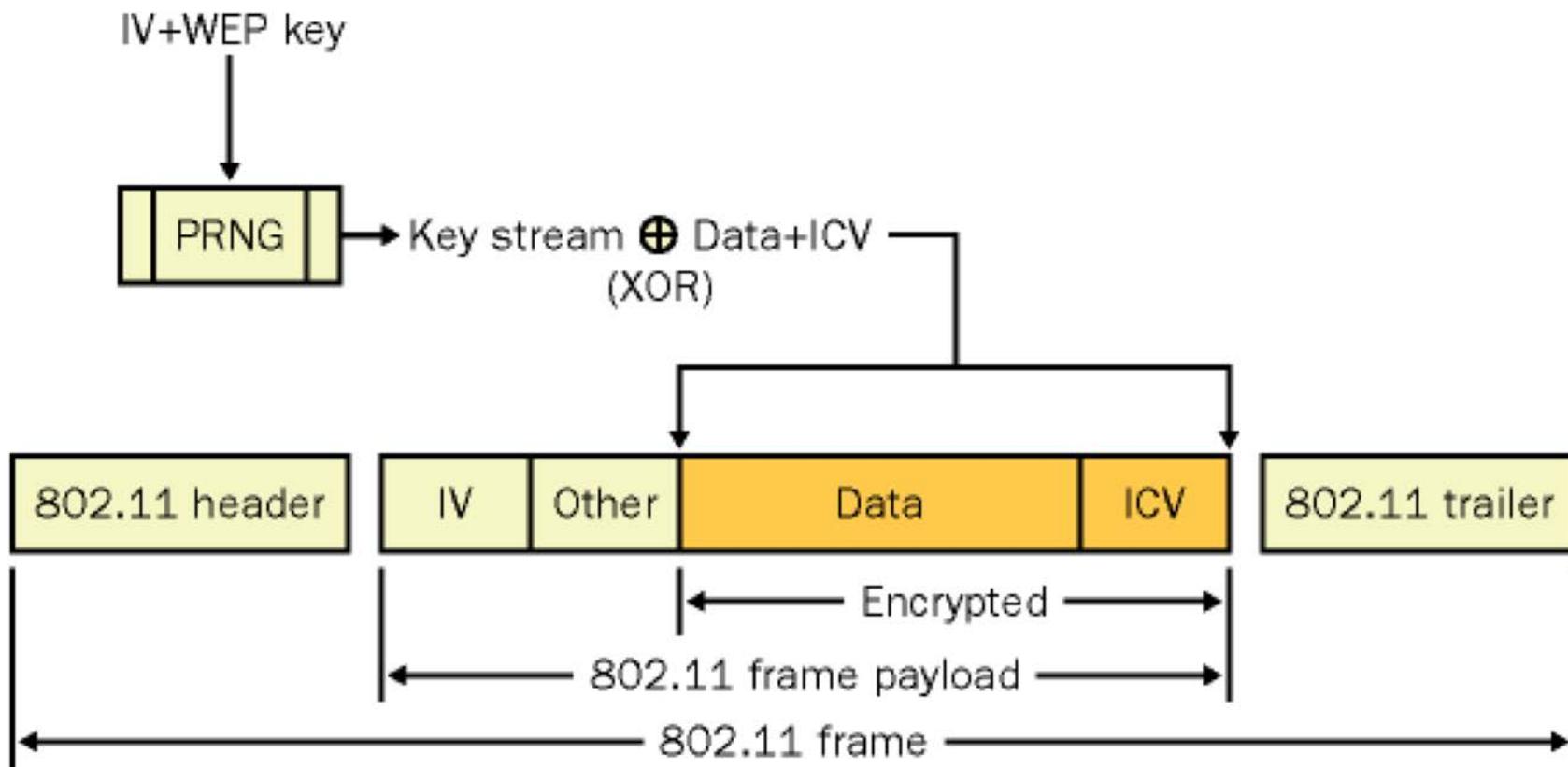
	Cableada	Inalámbrica
Ventajas	Robustez Ancho de banda Equipos baratos Fiables al contexto (ruido, vibración, interferencias u obstáculos)	Rapidez y bajo coste de instalación Bajo coste de mantenimiento Control local de estaciones e interacción con dispositivos, favoreciendo la movilidad Conexión para múltiples usuarios/dispositivos
Inconvenientes	Coste de mantenimiento Vulnerabilidad a amenazas físicas (principalmente) Dificultad para control en local	Vulnerables a múltiples tipos de amenazas No fiable para contextos inestables, ruidosos o con altas interferencias Coexistencia para interactuar varias redes inalámbricas

- En lo que a **seguridad** se refiere, hay dos características importantes que distinguen las LAN cableadas y las inalámbricas:
 - en una LAN cableada hay una forma intrínseca de autenticación de las estaciones porque están directamente interconectadas a esa red
 - una LAN cableada proporciona un cierto grado de privacidad porque la recepción de los datos está limitada a las estaciones conectadas a esa red
- Estas diferencias ponen de manifiesto la necesidad de servicios y mecanismos de seguridad más robustos en las LAN inalámbricas
- La especificación original de IEEE 802.11 incluía un conjunto de características de privacidad y autenticación para la privacidad; entre ellas, definió el protocolo **WEP (Wired Equivalent Privacy)**

- **WEP (Wired Equivalent Privacy)**

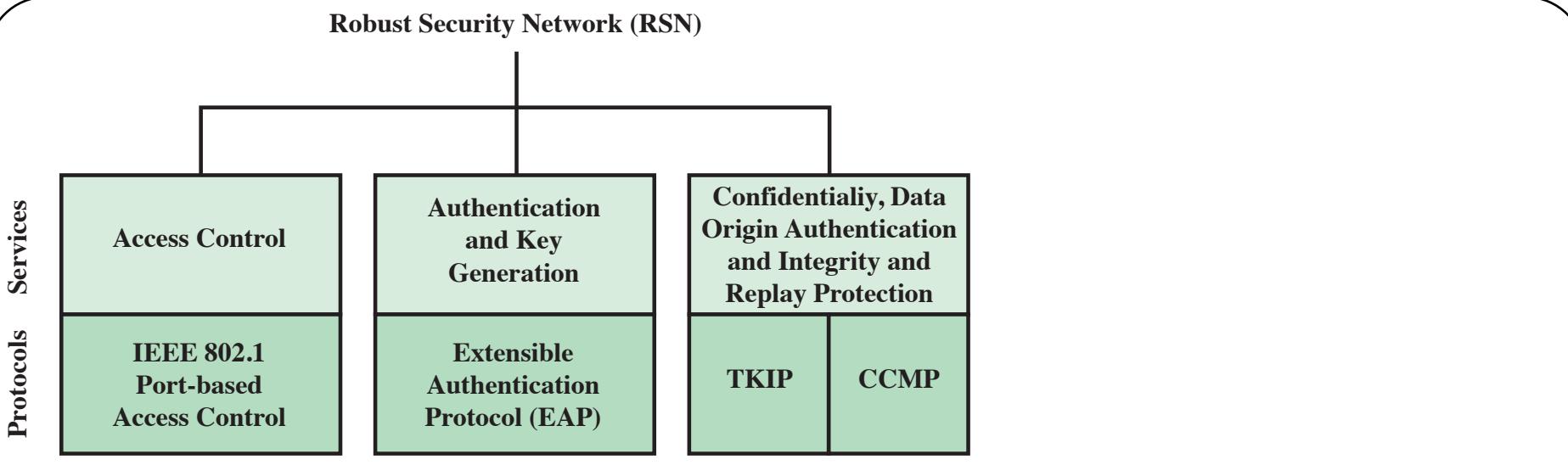
- El protocolo WEP se describió con el objetivo de proporcionar unos niveles de seguridad y privacidad comparables al de las LAN cableadas
 - limitado a la comunicación entre la estación y el punto de acceso
- se basa en la especificación de una clave de 64 bits que comparten los dispositivos de la red
 - de esos 64 bits, 40 corresponden a la clave secreta, y 24 al vector de inicialización (IV)
- WEP se basa en el algoritmo de cifrado RC4 (que es un algoritmo en flujo)
- WEP tiene varias vulnerabilidades:
 - Uso de IV débiles, que posibilitan que, a partir de un número de paquetes cifrados, el atacante puede recuperar la clave secreta
 - Reutilización de IVs, debido a la corta longitud del IV y su concatenación con la clave secreta

ICV: Integrity Check Value -- checksum

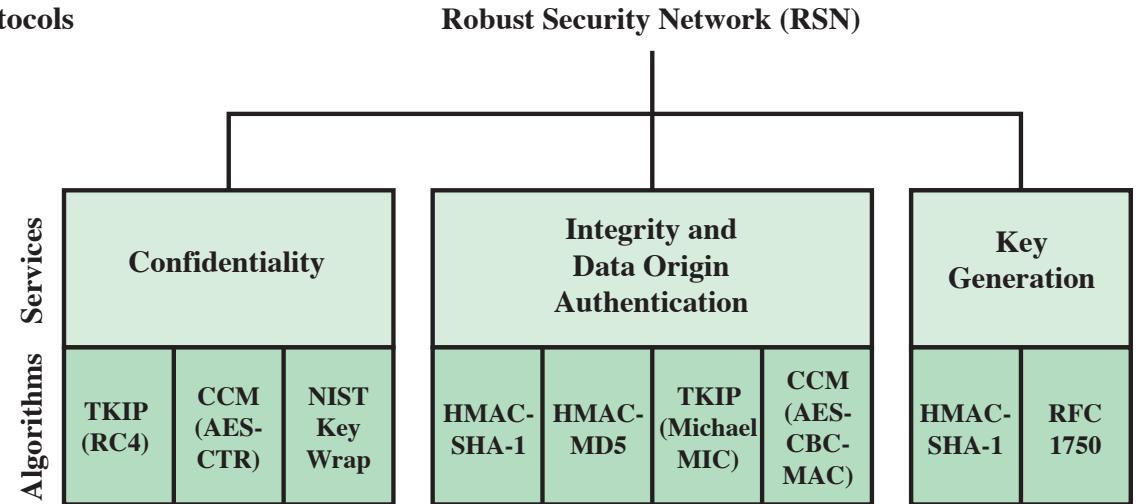


- Ante esa y otras debilidades iniciales, IEEE introdujo una serie de mejoras:
 - inicialmente, uso de **TKIP (Temporal Key Integrity Protocol)**
 - se basa también en la utilización del algoritmo RC4, pero genera nuevas claves de cifrado con cierta periodicidad, por lo que elimina el problema de los IV débiles
 - además, usa IVs de 48 bits (en lugar de 24 bits como en el caso de WEP)
 - posteriormente, uso de AES como algoritmo de cifrado
 - adopción del protocolo de autenticación 802.1X (desarrollado inicialmente para LAN cableadas)
- Todas estas mejoras de seguridad quedan recogidas e integradas dentro de la versión **IEEE 802.11i**
 - y, por lo tanto, en el protocolo **WPA (Wi-Fi Protected Access)**

- El 802.11i define los siguientes servicios:
 - **Autenticación**
 - se utiliza un protocolo entre un usuario y un servidor de autenticación, proporcionando autenticación mutua y generando claves temporales que se usan entre el cliente y el AP
 - **Control de acceso**
 - incluye el uso de la función de autenticación, el enrutado apropiado de los mensajes, y facilita el intercambio de clave
 - Puede trabajar con distintos protocolos de autenticación
 - **Confidencialidad con integridad de mensaje**
 - los datos se cifran con una función MAC que asegura que los datos no han sido alterados



(a) Services and Protocols



(b) Cryptographic Algorithms

- | | |
|---------|---|
| CBC-MAC | = Cipher Block Chaining Message Authentication Code (MAC) |
| CCM | = Counter Mode with Cipher Block Chaining Message Authentication Code |
| CCMP | = Counter Mode with Cipher Block Chaining MAC Protocol |
| TKIP | = Temporal Key Integrity Protocol |

- Las distintas operaciones de seguridad en el 802.11i se distribuyen entre 5 fases distintas
 1. Descubrimiento
 - el AP utiliza mensajes llamados *beacons* y *probe responses* para anunciar su política de seguridad
 - la estación los utiliza para identificar al AP y se asocia con él seleccionando un *cipher suite* y un mecanismo de autenticación
 2. Autenticación
 - la estación y el servidor de autenticación (AS) verifican la identidad del otro
 - hasta que la autenticación no ha finalizado, el AP bloquea cualquier tráfico entre la estación y el AS, salvo que el tráfico esté relacionado con el propio proceso de autenticación

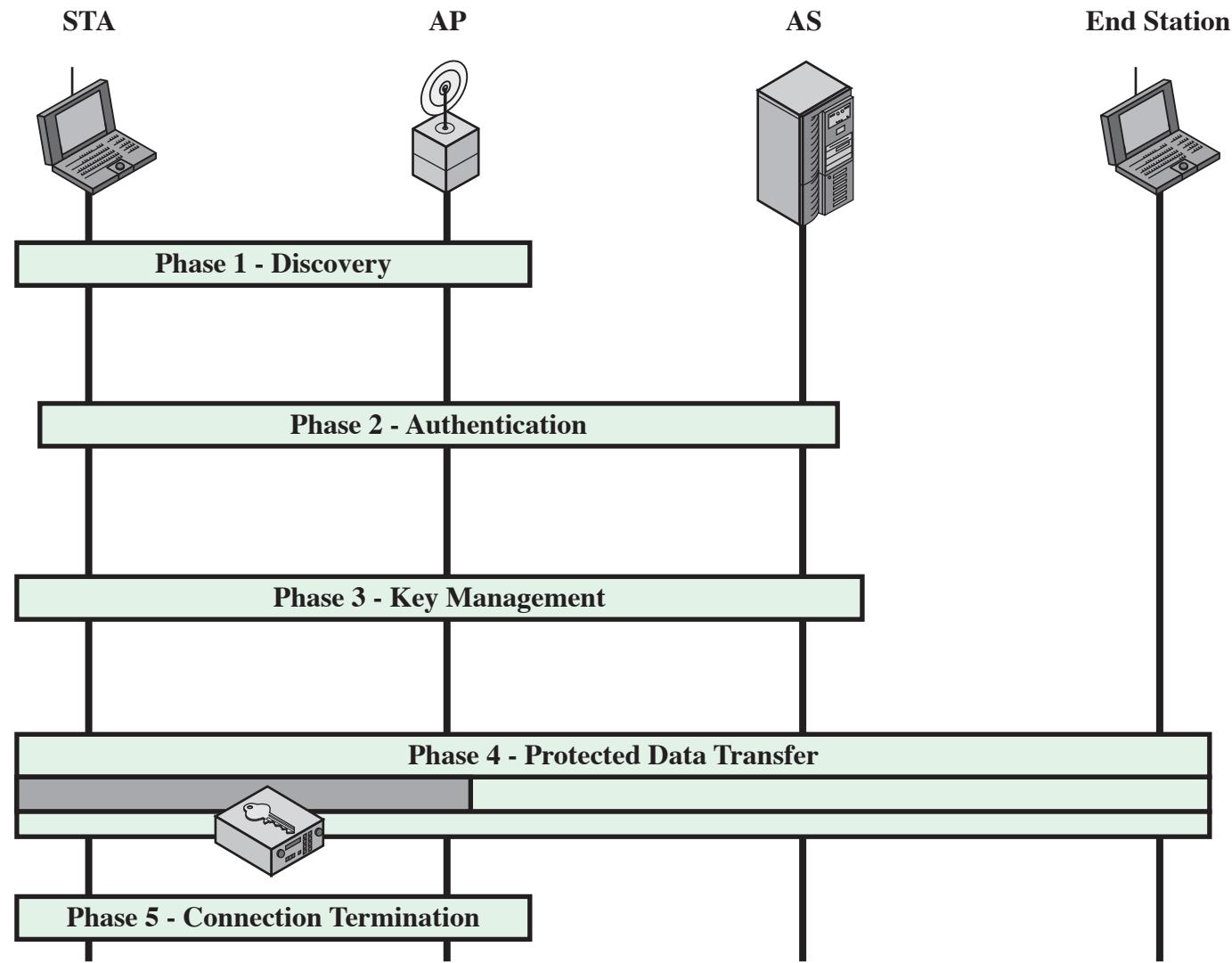
3. Generación y distribución de claves

- el AP y la estación realizan diferentes operaciones que generan claves criptográficas y que se almacenan en los propios AP y estación

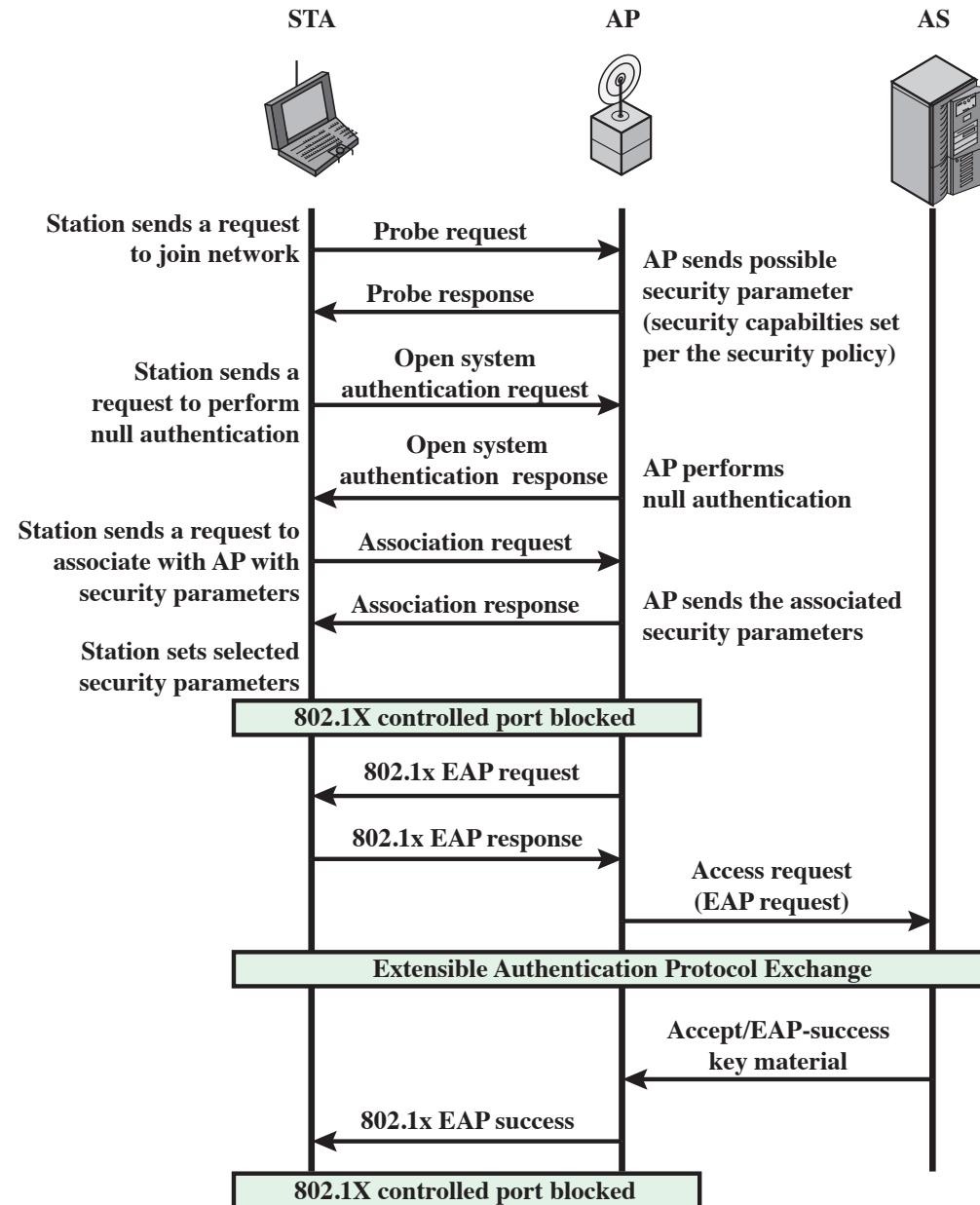
4. Transferencia segura de datos

- la estación origen y la estación final intercambian datos a través del AP pero la transferencia sólo se realiza de forma segura (cifrada) entre la estación de origen y el AP

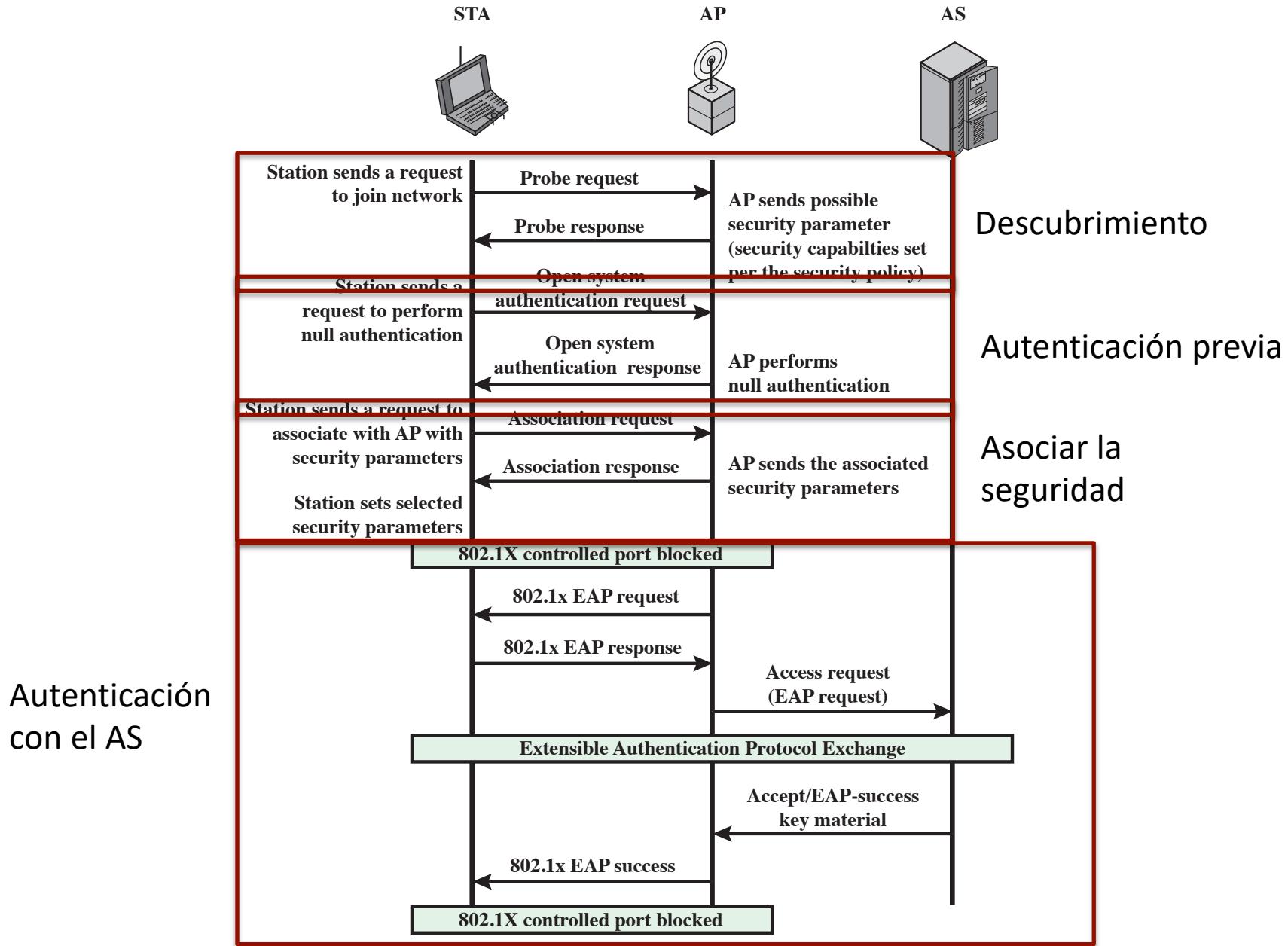
5. Finalización de la conexión



- Detalles de la fase de autenticación:
 - permite la autenticación mutua entre la estación y un servidor de autenticación (AS), como por ejemplo, un servidor RADIUS
 - la autenticación está diseñada para permitir sólo a estaciones autorizadas el uso de la red y garantizarles que están comunicando con una red legítima
 - el protocolo de autenticación utilizado es el **EAP (Extensible Authentication Protocol)**, definido en el estándar 802.1X
 - este estándar define los términos *suplicant*, *authenticator* y *authentication server* para nombrar a la estación, AP y AS, respectivamente
 - el authentication server puede ser un dispositivo por sí mismo, o puede ejecutarse dentro del propio authenticator

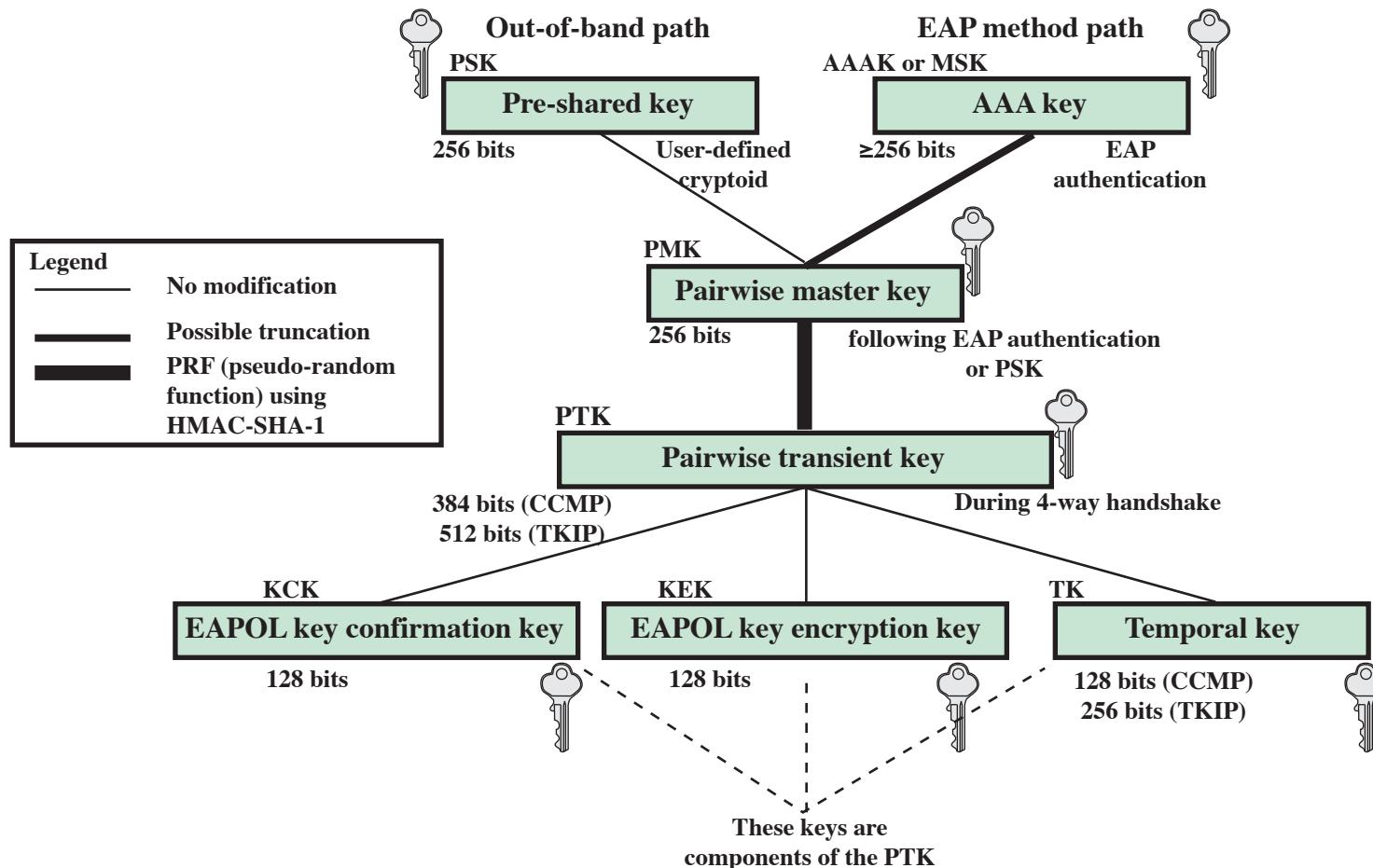


**IEEE 802.11i Phases of Operation:
Capability Discovery, Authentication, and Association**



**IEEE 802.11i Phases of Operation:
Capability Discovery, Authentication, and Association**

- Detalles de la fase de generación de claves:



- Detalles de la fase de generación de claves:
 - Se basa en la composición global de varias claves organizadas según una jerarquía
 - Cuando la autenticación es exitosa, se crean claves temporales de sesión
 - La clave PMK (Pairwise Master Key) depende del método de autenticación
 - » Si se aplica una PSK (Pre-Shared Key), entonces: PMK = PSK, cuyo valor puede derivar de una frase secreta de 8-23 caracteres, o una cadena de 256-bit. Solución adecuada para entornos pequeños sin servidor configurado
 - » Si se usa un servidor SA, entonces PMK puede derivar de la MK (Master Key) del SA
 - Con la PMK se genera una clave temporal para cifrado denominada PTK (Pairwise Transient Key). Su longitud depende del protocolo de cifrado (TKIP-512, CCMP-384). La PTK está basado de varias claves dedicadas:
 - » KCK (Key Confirmation Key – 128 bits): usada para la autenticación de mensajes (MIC)
 - » KEK (Key Encryption Key – 128 bits): usada para garantizar la confidencialidad de los datos enviados
 - » TK (Temporary Key – 256/128 bits): clave para cifrar datos en modo TKIP o CCMP

REFERENCIAS BIBLIOGRÁFICAS

Bibliografía básica

- “*SSL and TLS: Theory and Practice*”
Rolf Oppliger
Artech House, 2009
- “*Demystifying the IPsec Puzzle*”
Sheila Frankel
Artech House, 2001
- “*Cryptography and Network Security: Principles and Practice*”
William Stallings
Prentice Hall, 2010 (5^a edición)
- “*Computer Security: Principles and Practice*”
William Stallings and Lawrie Brown
Prentice-Hall, 2011 (2^a edición)

Referencias

- RFC 5246, “*The Transport Layer Security (TLS) Protocol, Version 1.2*”, 2008
- RFC 6347, “*Datagram Transport Layer Security, Version 1.2*”, 2012
- RFC 1636, “*Report of IAB Workshop on Security in the Internet Architecture*”, 1994
- RFC 4301, “*Security Architecture for the Internet Protocol*”, 2005
- Guillaume Lehembre, Seguridad Wi-Fi – WEP, WPA y WPA2, hakin9 No 1/2006,
http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_ES.pdf