Tema 5: Seguridad en redes TCP/IP

Seguridad en la Capa de Transporte

Introducción

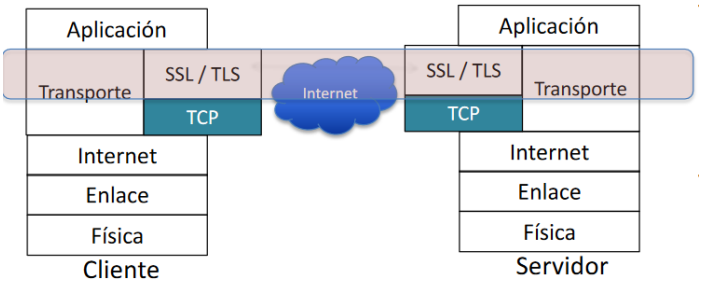
La expansión de las WWW en los 90 y su utilidad para realizar transacciones a nivel web plantean nuevos riesgos de seguridad:

- En el lado del **cliente** web.
- En el lado del **servidor** web.
- **La propia información** entre cliente y servidor.

	AMENAZAS	CONSECUENCIAS
Integridad	**Modificación** datos usuario, troyano navegador, modificación de mensaje en tránsito…	**Pérdida de la información**, afecta al dispositivo, desencadenamiento de otros ataques.
Confidencialidad	**Escuchas** en el canal de comunicaciones, robo información, robo configuración de red	**Pérdida de información y privacidad**
Denegación de servicio	**Interrumpir** procesos de usuario, interrupciones en el lado del cliente/servidor	Interrupción inaccesibilidad
Autenticación	**Suplantación** de identidad, falsificación de datos	Creer en **datos falsos** o entidades ilegítimas

La IEFT formó a mediados de los 90 un grupo de trabajo denominado ***Web Transaction Security*** **(WTS).** Su objetivo era: desarrollar los requisitos y las especificaciones para la **provisión de servicios de seguridad en transacciones web**. Este grupo se centró en el desarrollo de una solución en la **capa de aplicación** y diseño **SHTTP** (*Secure HyperText Transfer Protocol*), especificado en el RFC 2084 (*Considerations for Web Transaction Security*), RFC 2659 (*Security Extensions for HTML*) y RFC 2660 (*The Secure HyperText Transfer Protocol*).

Por otro lado, en las mismas fechas, los desarrolladores de Netscape abordaron el problema pero desde la **capa de transporte**, como una solución intermedia, ni en la capa alta ni en la baja. El resultado fue el protocolo **SSL** (*Secure Sockets Layer*), una subcapa entre la de aplicación y la de transporte. Más concretamente SSL se sitúa por encima de TCP dado que es orientado a la conexión y proporciona fiabilidad.



El objetivo de SSL es crear conexiones seguras y transmitir los datos en esas conexiones.

El protocolo **TLS** (*Transport Layer Security*) es una iniciativa de IEFT para estandarizar SSL, se definió por primera vez en 1999 (RFC 2246). Las diferencias entre TLS 1.0 y SSL 3.0 no son dramáticas, pero son significantes para imposibilitar la interoperabilidad entre TLS 1.0 y SSL 3.0.

Cuándo	Quién	Qué
A mediados de 1994	Netscape	SSL v1.0
A finales de 1994	Netscape	SSL v2.0
1995	Netscape	SSL v2.0
1996	Netscape	**SSL v3.0 (obsoleta)**
1999	IETF	TLS v1.0 (SSL v3.1) – RFC: 2246
2006	IETF	TLS v1.1 (SSL v3.2) – RFC: 4346
2008	IETF	TLS v1.2 (SSL v3.3) – RFC: 5246
2014	IETF	TLS v1.3 (draft 1) – (SSL v3.4) – RFC: 8446
2018	IETF	TLSv 1.3

SSL (*Secure Sockets Layer*)

El protocolo SSL es un protocolo **cliente-servidor** que proporciona los siguientes servicios de seguridad entre los puntos que se comunican:

- **Autenticación** de entidades y de origen de datos.
- **Confidencialidad** de la conexión.
- **Integridad** de la conexión.

Más concretamente SSL emplea:

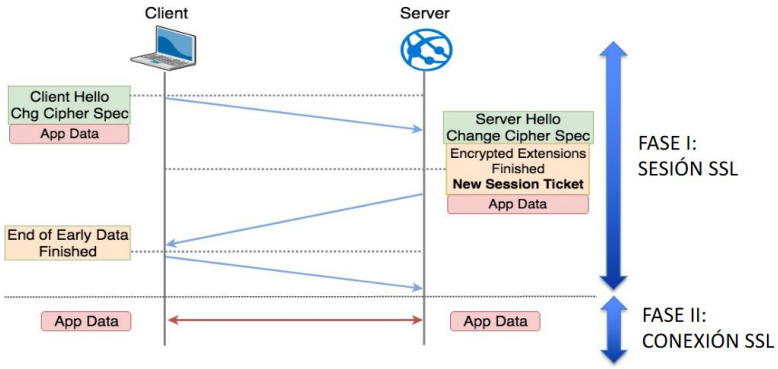
- **Criptografía simétrica**: para la autenticación de los datos (mensajes) y el cifrado.
- **Criptografía asimétrica:** para la autenticación de entidades y para el establecimiento de clave.
    - RSA, Diffie-Hellman y Fortezza KEA (hasta SSL 3.0).
    - A pesar de la criptografía asimétrica **no proporciona no-repudio**.

Una ventaja del protocolo es que es **independiente del protocolo de la capa de aplicación**, es decir, cualquier protocolo de aplicación basado en TCP se puede beneficiar de SSL (este le da los servicios de seguridad mencionados).

Protocolo	Descripción	Puerto
https	HTTP sobre SSL/TLS	443
ldaps	LDAP sobre SSL/TLS	636
ftps-data	FTP Data sobre SSL/TLS	989
ftps	FTP Control sobre SSL/TLS	990
telnets	Telnet sobre SSL/TLS	992
Impas	IMAP4 sobre SSL/TLS	993
Pop3s	POP3 sobre SSL/TLS	995
…	…	…

El protocolo SSL emplea los siguientes conceptos:

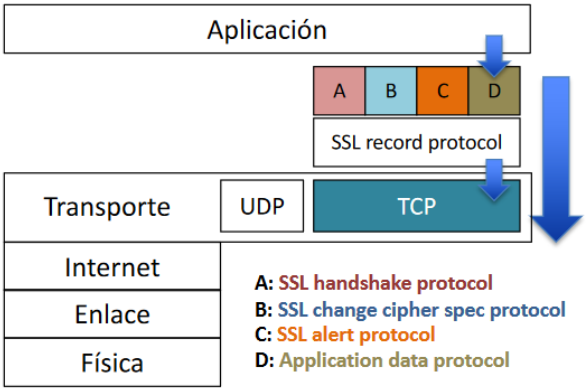
- **Sesión SSL**: asociación entre el cliente y el servidor en la que se negocian los parámetros de seguridad para todas las conexiones de esa sesión.
- **Conexión SSL:** realización de la transmisión de datos entre el cliente y el servidor, protegida criptográficamente según lo anterior.



El ámbito de la funcionalidad de SSL es doble, como se desprende de lo anterior:

1. Establecer una **conexión segura** (confidencial y autenticidad), entre los puntos que se comunican.
2. Utilizar esa conexión para **transmitir de forma segura los datos** del nivel de aplicación entre emisor-receptor. Esta transmisión requiere:
    a. Dividir los datos en fragmentos más manejables.
    b. Procesarlos de forma individual, cada fragmento se denomina SSL *record*.

Para llevar a cabo esta doble funcionalidad, SSL consta de dos subcapas y varios subprotocolos.



La **subcapa alta** contiene:

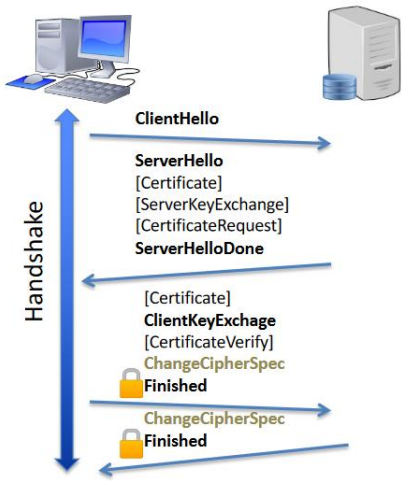
1. ***SSL Handshake Protocol***: permite que los puntos de comunicación se autentiquen mutuamente, y que además, negocien un ***cipher suite*** y (opcionalmente) método de **compresión**.
2. ***SSL Change Cipher Spec Protocol:*** permite a los puntos de comunicación activar el *cipher suite*.
3. ***SSL Alert Protocol***: permite a los puntos de comunicación indicar posibles problemas potenciales e intercambiar los correspondientes mensajes de alerta.
4. ***SSL Application Data Protocol***: es el propio protocolo de la capa de aplicación (ejemplo: HTTP) y alimenta al *SSL Record Protocol.*

La subcapa baja contiene:

1. ***SSL Record Protocol***: fragmenta los datos de la capa de aplicación y los procesa de forma individual.

SSL consta de las siguientes fases:

- **FASE 1:** se establecen los parámetros de seguridad, versión del protocolo, la sesión de ID, el suite de cifrado, el método de compresión (opcional) y un número aleatorio.
- **FASE 2:** el servidor puede enviar un certificado, intercambia los parámetros necesarios para gestionar la clave secreta y puede solicitar un certificado del cliente.
- **FASE 3:** el cliente envía el certificado si es solicitado y los parámetros de seguridad necesarios para computar clave de sesión. Si el cliente envía certificado, entonces necesita enviar un certificado firmado para la verificación de la entidad origen.
- **FASE 4:** se establece el suite de cifrado y se termina el proceso de handshake.



Sin embargo, para proteger la conexión necesitamos computar (1) **clave de sesión**, (2) **clave para MAC**, (3) **IV** para modo de operación. Estos parámetros se calculan en función de varios componentes de seguridad: (1) **semilla**, (2) valor aleatorio nonce, (3) una función pseudoaleatoria (PRF *PseudoRandom Runction*).

Intercambio de claves

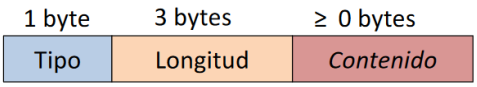
Para el intercambio de claves, SSL/TLS usan DHE (*Diffie-Hellman Efímero*). Tanto el cliente como el servidor **generan sus valores secretos en cada negociación**, en lugar de usar valores estáticos para el intercambio. Con esto se consigue el **PFS** (*Perfect Forward Secrecy*) en la creación del secreto compartido. Evita el riesgo de MiTM (*Man in The Middle*); si la clave privada de RSA (servidor) se filtra o hay un MiTM en las negociaciones DH entre las dos entidades, el MiTM puede derivar las claves de sesión que se establezcan entre ambas entidades legítimas.

SSL Handshake Protocol

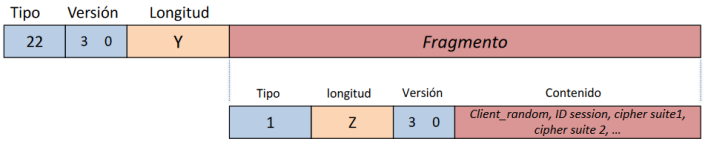
Se utiliza antes de transmitir ningún dato en la capa de aplicación. Es la parte más compleja de SSL poque permite a servidor y cliente:

- **Autenticarse** mutuamente.
- Negociar un **algoritmo de cifrado** y una función **MAC**.
- Así como las **claves** para proteger los datos del SSL record.

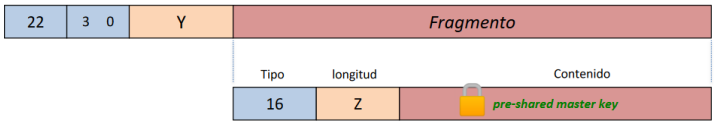
Consta de mensajes con el formato:



- *Tipo*: indica uno de los posibles 10 mensajes.
    - (Tipo 0) *hello_request*.
    - (Tipo 1) *client_hello*.



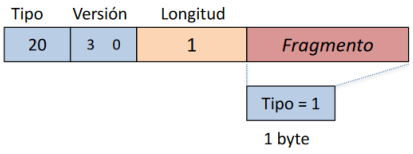
    - (Tipo 2) *server_hello.*
    - (Tipo 11) *certificate.*
    - (Tipo 12) *server_key_exchange*.
    - (Tipo 13) *certificate_request*.
    - (Tipo 14) *server_hello_done*.
    - (Tipo 16) *client_key_exchange*.



    - (Tipo 15) *certificate_verify*.
    - (Tipo 20) *finished*.
- *Longitud*: longitud del mensaje en bytes.
- *Contenido*: parámetros asociados al mensaje

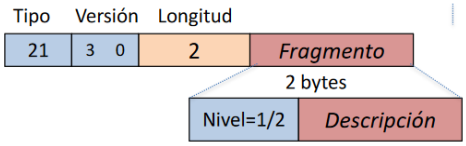
SSL Change Cipher Spec Protocol

Protocolo **muy simple**, con un único mensaje de un solo byte con valor 1 que permite activar el cipher suite.



SSL Alert Protocol

Se utiliza para **comunicar** al otro punto de comunicación las alertas relacionadas con SSL, y cada mensaje de este proceso consta de 2 bytes.

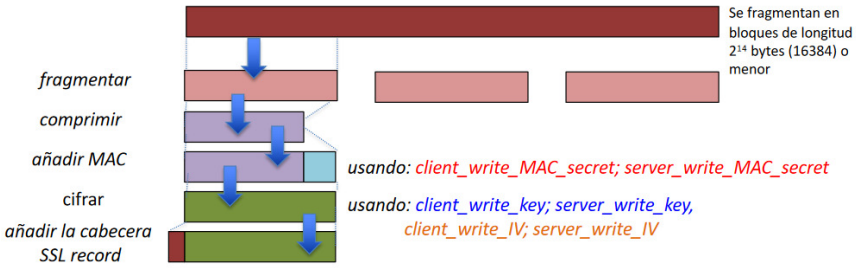


- **Primer byte**: toma el valor 1 (warning) o 2 (fatal) para informar de la severidad.
    - Fatal → termina la conexión.
    - Otras conexiones de la misma sesión pueden continuar pero no se producirán nuevas conexiones dentro de la misma sesión.

- **Segundo byte:** contiene código que indica la alerta específica. Ejemplos: *unexpected_message, bad_record_mac, decompression_failure…*

SSL Record Protocol

Toma los datos de la subcapa alta, los **fragmenta** en bloques manejables, los **comprime** (opcional), **añade MAC**, **cifra** y **añade cabecera**. El resultado final se transmite en fragmento TCP. En el destino los datos recibidos son descifrados, verificados, descomprimidos y reensamblados antes de entregarlos a la capa de aplicación.



TLS v1.2

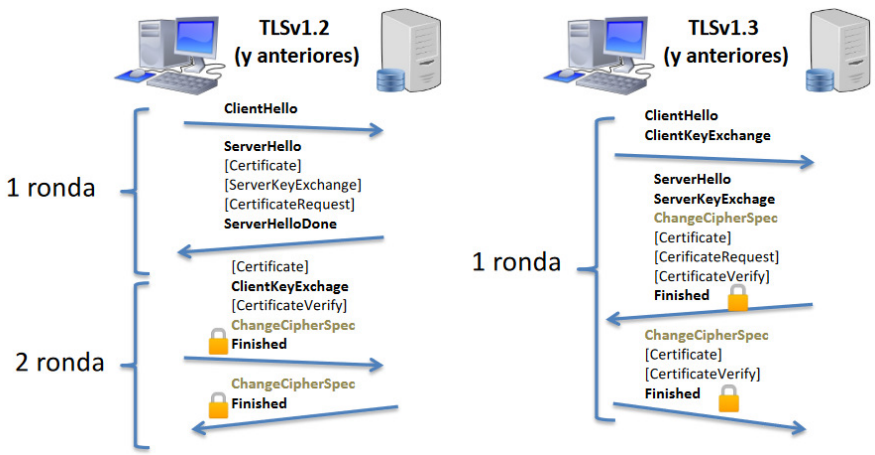
TLS v1.2 introduce cambios muy significativos:

- Handhsake:
  - El *master key* se computa con SHA-256 (en lugar de MD-5 y SHA-1).
  - Permite incluir una lista de **extensiones** en los mensajes *ClientHello/ServerHello* y detalles sobre los certificados, parámetros de seguridad, autorizaciones, tipos de certificados…
- Cipher suite:
  - Quita DES, IDEA añade AES.
  - Añade **criptografía de clave pública** basada en curvas elípticas con ECDHE (negociación claves).
  - Introduce el concepto "Authenticated Encryption with Addition Data" (AEAD).
    - AES-CBC-MAC / AES-CCM.
    - AES-GCM (Galois-Counter Mode).

TLS v1.3

TLS v1.3 introduce cambios muy significativos en términos de rendimiento y seguridad:

- Un único Round-Trip Time (RTT)

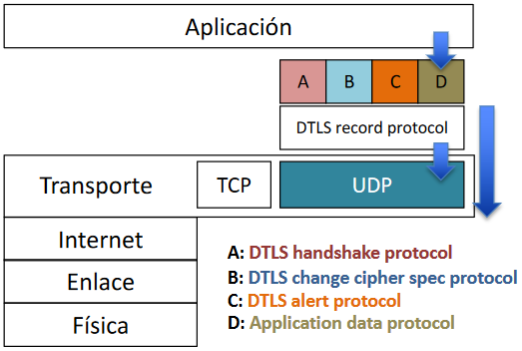


- 0-RTT para reconexión por usar las credenciales de seguridad de la sesión anterior - PSK (*Pre-shared Master Key*).
- Prevalece sesiones del tipo Perfect Forward Secrecy - DHE-RSA / ECDHE.
- Reduce el número de modos de operación soportados, limitándose a CBC y AEAD: GCM y CCM.

DTLS (*Datagram Transport Layer Security*)

Es conveniente comentar que existe un protocolo llamado DTLS definido en el RFC 6347. Se utiliza para protocolos **basados en datagramas,** los que se ejecutan por **UDP**. Se creó en 2006, aunque la última versión es de enero de 2012. Está tomando un papel relevante en escenarios que se requiera comunicación en tiempo real o restringidos (IoT).

El funcionamiento es similar al de TLS:



Tiene la diferencia de que la comunicación va sobre UDP. Esto significa que los datagramas son transmitidos de forma no fiable (puede haber pérdidas o ensamblado no ordenado). Para evitar esto se podría:

- Incluir mecanismos para controlar el flujo de información. Un explícito número de secuencia en cada DTLS record.
- Evitar cualquier dependencia con los mecanismos de protección, no usar algoritmos de cifrado en flujo, usar CBC con estado.

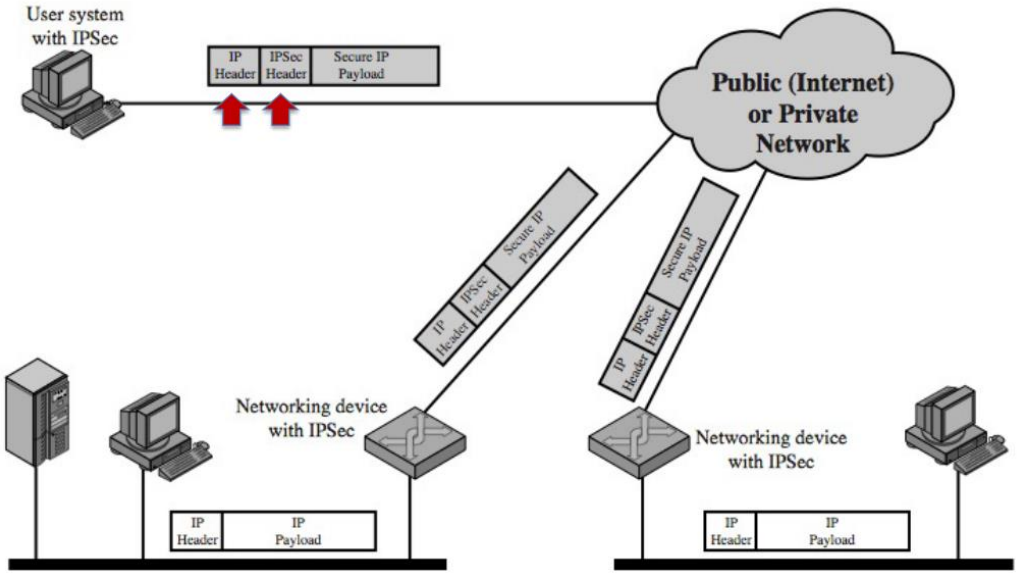
Seguridad en la Capa de Internet (red)

En 1994, la *Internet Architecture Board* (IAB) publicó un informe titulado "Security in the Internet Architecture" (RFC 1636). En este informe se identificaba la necesidad de proporcionar **seguridad a la infraestructura de red**, aconsejando la incorporación de mecanismos de **cifrado** y **autenticación** para la siguiente versión de IP, IPv6.

A partir de ese momento, se elaboraron, bajo el nombre de **IPSec** (RFC 4301) las especificaciones y las funcionalidades de seguridad en la capa de Internet para el modelo TCP/IP. No solo teniendo en cuenta IPv6, sino que también sirvieran para IPv4.

Implementando la seguridad al nivel de IP, una empresa garantiza la protección de todas sus aplicaciones, necesiten estas seguridad o no. Por ello se pueden usar en múltiples escenarios:

- Conectividad segura entre sucursales a través de internet.
- Acceso remoto seguro sobre internet.
- Establecimiento de conectividad extranet e intranet con socios.
- Aplicaciones de comercio electrónico…



Servicios de IPSec

La **seguridad en IPSec** se centra en:

- **Autenticación + Integridad** (MAC)
- **Confidencialidad**
- **Intercambio de claves entre los puntos**.

Como se ha comentado **es transparente a las aplicaciones**, al trabajar por debajo del nivel de transporte. Por lo tanto, es transparente a los usuarios finales, lo que implica (1) no es necesario formar a los usuarios sobre el uso de mecanismos de seguridad, (2) no hace falta gestión de claves.

IPSec no proporciona:

- Servicios de **no-repudio**, al igual que SSL/TLS.
- Protección frente a ataques DoS, aunque proporciona una forma de protección ante ataques de repetición.

Protocolos de IPSec

Para la comunicación entre dos puntos utiliza los siguientes protocolos:

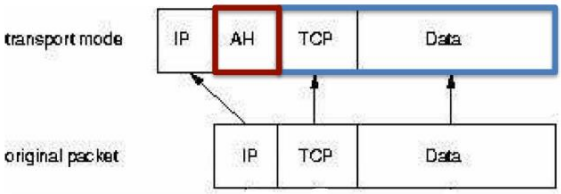
- **ESP** (*Encapsulating Security Payload*).
    - Cabecera para **confidencialidad, integridad y autenticación** del origen de datos.
    - Incluye número de secuencia para protección de ataques **de repetición**.
    - Protección parcial ante análisis de tráfico (modo túnel)
- **AH** (*Authentication Header*)
    - Cabecera para **integridad y autenticación** del origen de datos.
    - Incluye número de secuencia para protección de ataques **de repetición**.
- **IKE** (*Internet Key Exchange*)
    - Protocolo para **generar y distribuir** claves criptográficas para ESP y AH.
    - Autentica la identidad del sistema remoto.

Antes de que dos puntos se comuniquen de forma segura tienen que acordar qué parámetros de seguridad se van a aplicar.

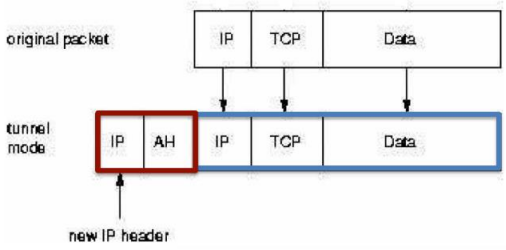
Modos de IPSec

Presenta dos modos:

- Modo transporte:
    - Se usa normalmente para comunicaciones punto a punto **entre dos hosts**.
    - Proporciona protección a la **carga útil del paquete IP** (*payload*), es decir, protocolos de capa superior: TCP, UDP, ICMP.



- Modo túnel:
    - Se suele usar cuando los puntos a comunicar son **gateways** de seguridad o bien **routers**.
    - Proporcionan protección al **paquete IP**.
    - Encapsula el datagrama IP dentro de un nuevo datagrama IP que emplea IPSec.



Como puede verse la protección a **todo el paquete IP** puede lograrse:

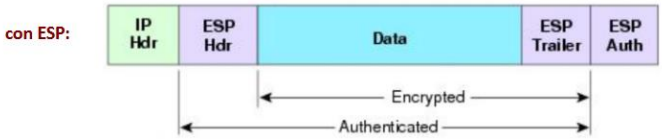
- Añadiendo las cabeceras AH o ESP al paquete IP original.
- Creando una cabecera IP nueva.

De esta forma el paquete IP original se “encapsula” y viaja por el túnel sin que ninguno de los routers intermedios pueda saber **ni el origen ni el destino final de los datos**.

Modo transporte

En este modo de uso:

- Si se utiliza ESP: se cifra y opcionalmente autentica el payload, pero no la cabecera.



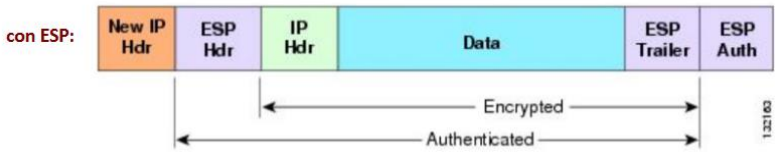
- Si se utiliza AH: se autentica el payload y algunas porciones de la cabecera.

La forma de enlazar la cabecera IP con la cabecera ESP/AH es estableciendo el campo *proto* de la primera al código del protocolo ESP (50) o AH (51).

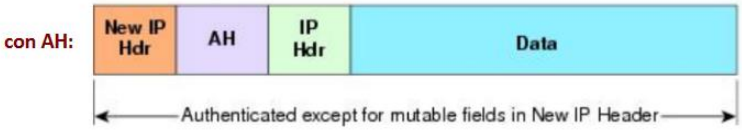
Modo túnel

En este modo:

- Si se usa ESP: se cifra y opcionalmente autentica todo el paquete IP original (paquete interno), incluyendo la cabecera de ese paquete original.



- Si se usa AH: se autentica todo el paquete original y algunas partes de la cabecera externa.



Asociaciones de seguridad

Con IPSec se tienen que configurar varios aspectos:

1. El **modo de integridad** de los mensajes por incluir en la cabecera del protocolo IPSec, el HMAC.
2. El **uso general de algoritmos de cifrado** estándar.
3. Controlar un tipo de ataque de DoS basado en **replay** por aplicar un número de secuencia única de paquetes.
    a. *Sequence number*
    b. Solo se aceptan paquetes que tienen un número actual de secuencia o posterior, las anteriores se descartan.
4. El modo de **encapsular y desencapsular** paquetes IPSec,
    a. Para ello, se requiere el uso de algún mecanismo que almacene las claves secretas, los algoritmos de cifrado y autenticación, y las direcciones IP involucradas en la comunicación.

Esto último constituye las **Asociaciones de seguridad, SA (*Security Associations*).** Una SA define:

1. **Dirección IP** origen y destino.

2. **Algoritmo y clave secreta** empleados para AH y ESP, a veces compresión.
3. El índice de parámetro de seguridad (**SPI- *Security Parameter Index***): número de 32 bits que identifica la asociación de seguridad.
4. **Solo protege un sentido:** el emisor y el receptor deben aplicar la misma SEA pero teniendo en cuenta el destino y el origen.

Las SAs se almacenan en una **base de datos de asociaciones de seguridad (SAD)**. Para cada entrada en la SAD existen varios campos:

- *Security Parameter Index* (SPI): 32 bits para identificar a SA.
- *AH Information*: algoritmo de autenticación, claves y otros parámetros relacionados con AH.
- *ESP Information*: algoritmo de cifrado y autenticación, claves y otros parámetros relacionados con ESP.
- *Lifetime of the SA:* un intervalo o contador después del cual habrá que reemplazar la SA.
- El tipo de modo: túnel o transporte.

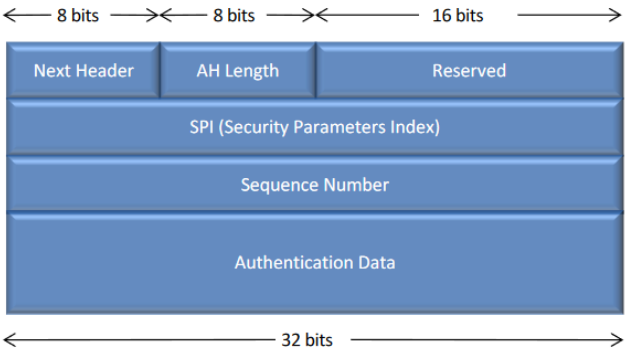
Política de seguridad

Una SA solo identifica "el modo en que se protegerá el dato IPSec", para definir "el modo en que viajará el tráfico entre dos puntos" se requiere una **política de seguridad SP** (*Security Policy*) que se almacena en una SPD (*Security Policy Database*). Un SP define:

- Direcciones de origen y destino a proteger.
  - En modo transporte serán las mismas direcciones que aquellas definidas en la SA.
  - En modo túnel no tienen que ser las mismas.
- Los protocolos y puertos a proteger
  - Algunas implementaciones no permiten la definición de protocolos específicos a proteger. En este caso se protege todo el tráfico.
- El modo de protección: túnel o transporte.

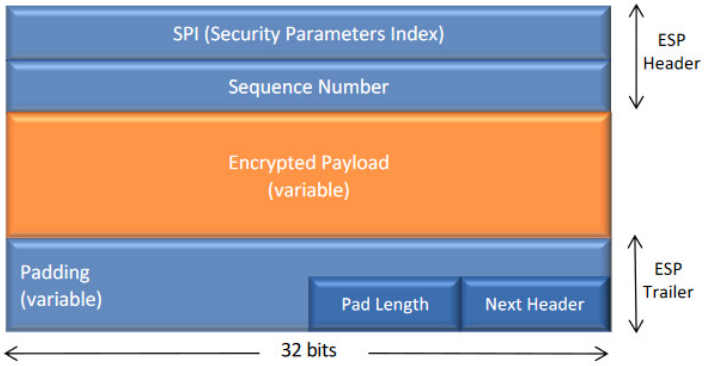
Cabeceras AH y ESP

Authentication Header (AH)



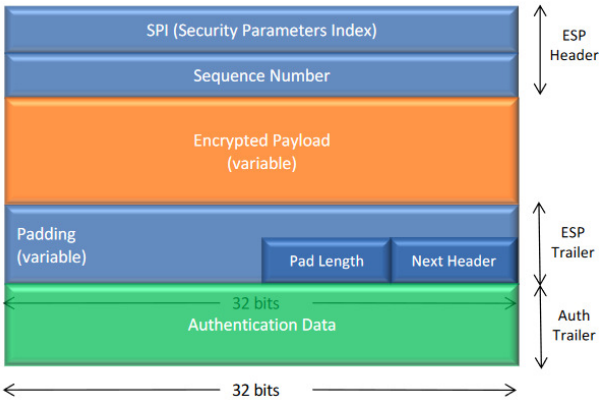
- *Next header:* cabecera inmediatamente posterior.
- *AH length:* longitud de la cabecera en palabras de 32 bits, menos palabras.
- *SPI* (*Secuirty Parameters Index*): identifica una SA.
- *Sequence number*: contador para evitar ataques replay.
- *Authentication Data*: contiene el valor de comprobación e integridad MAC para este paquete.

Encapsulating Security Payload (ESP)



- *SPI*: identifica una SA.
- *Sequence number:* evitar ataques replay.
- *Encrypted Payload:* segmento del nivel de transporte (modo transporte) o paquete IP (modo túnel) protegido por medio de cifrado.
- *Padding*: espacio adicional requerido incluido porque los algoritmos de encriptación basados en bloques pueden requerir espacios diferentes.
- *Pad length*: longitud del *padding*.
- *Next header*: guarda el tipo de la siguiente cabecera.

SI ESP incluye cifrado + autenticación del dato + integridad la cabecera toma la forma:



Las distintas opciones de qué cabeceras usar en función de los servicios que se quiere proporcionar son:

	AH	ESP (encryption only)	ESP (encryption plus authentication)
Connectionless integrity	✔		✔
Data origin authentication	✔		✔
Rejection of replayed packets	✔	✔	✔
Confidentiality		✔	✔
Limited traffic flow confidentiality		✔	✔

IKE

Como hemos podido comprobar, cuando no existe SA hay que negociarla; de eso se encarga el protocolo IKE (*Internet Key Exchange*). IKE se encarga de la:

- Autenticación de las partes de la comunicación.

- El establecimiento de la clave secreta.

Utiliza:

- Certificados X.509 para la autenticación
- Algoritmo de Diffie-Hellman para establecer clave secreta.

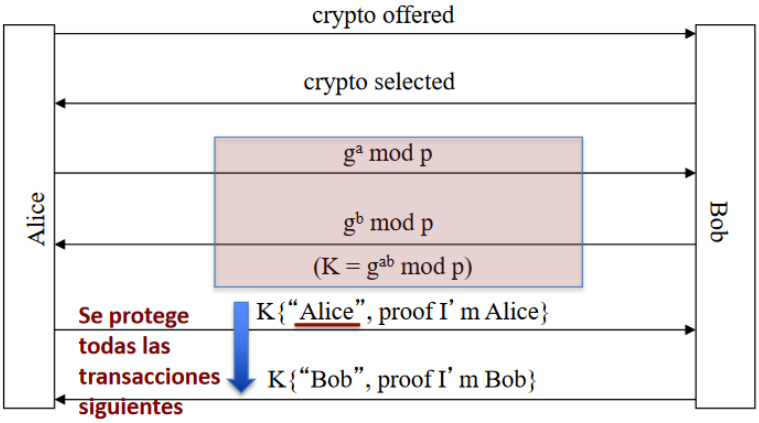
Se basa en los protocolos:

- Oakley (*Key Exchange Protocol*)
- ISAKMP (*Internet Security Association and Key Management Protocol*).

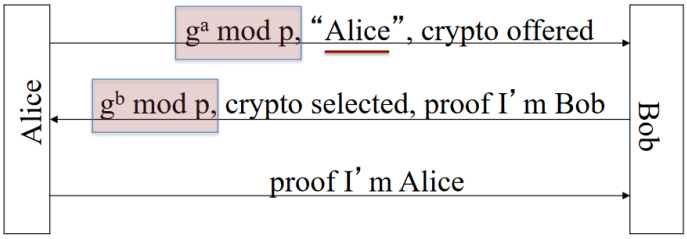
El funcionamiento está compuesto por dos fases:

- **FASE 1**: **establece una SA ISAKMP** previa a la SA de IPSec.
    - La autenticación de las partes suele basarse en claves compartidas, claves RSA y certificados X.509.
    - Esta fase soporta dos modos: principal y agresivo (simple y la mitad de mensajes, pero no soporta la protección completa de identidades).
- **FASE 2**: el neuvo SA ISAKMP es empleado para negociar y establecer los **SAs de IPSec**.
    - En esta fase IKE intercambia propuestas de SAs.
        - Negocia asociaciones de seguridad basándose en el ISAKMP SA inicial
        - Establece clave de sesión.
    - Las claves de las SAs se derivan de
        - las claves de la primera fase, los nonce y los SPI o usando un nuevo Diffie-Hellman.

El modo principal consiste en los siguiente:



Y el modo agresivo se basa en:



Firewalls en redes

Definición

Un **cortafuegos** (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 (internet) no confiable, llamada **zona de riesgo**.

Zonas desmilitarizadas (De-Militarizeed Zones, DMZ) 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 esta forma se 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*.

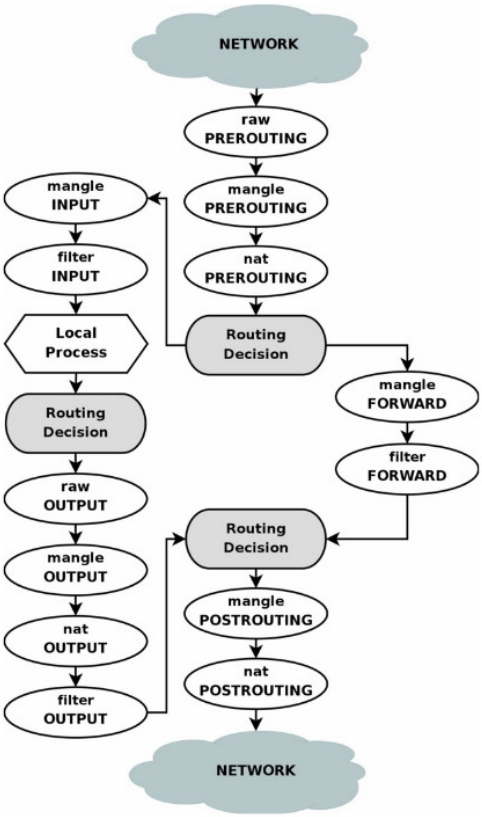
- Tipos de reglas: añadir, borrar o crear reglas para tráfico entrante/saliente.
- Objetivos:
    - Habilitar el acceso a puertos específicos y a determinadas IPs, permitiendo no solo el acceso TCP, sino UDP, ICMP…
    - Denegar el acceso desde redes externas (o internas) a puertos específicos.
    - Enmascarar tráfico de red local hacia redes públicas externas.

La sintaxis de las reglas suele ser: *iptables -A <chain> -j <target>*

- *<chain>*: define una cadena que puede ser del tipo INPUT, OUTPUT, 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, qué hacer, ACCEPT o DROP
- *-p <protocol>*: especifica el protocolo del puerto, ejemplo: tcp, udp…
- *--dport <nport>*: especifica el número de puerto destino.
- *--sport <nport>*: especifica el número de puerto origen.
- *-s <ip>*: especifica IP origen (se permite especificar máscara).
- *-d <ip>*: especifica IP destino (se permite especificar máscara).
- *-P <chain> <target>*: para establecer políticas por defecto.

La comunicación hacia o desde redes públicas requiere:

- POSTROUTING: interior → exterior.
    - Realizar un proceso de **enmascaramiento** para poder trabajar con NAT.
    - Primero *FORWARD/OUTPUT*, luego *POSTROUTING*.
- PREROUTING: exterior → interior.
    - Primero *PREROUTING,* luego *FORWARD/INPUT*.



Además, un firewall debe controlar el problema del *three-way-handshake* en TCP. Una máquina que desea contactar a otra envía un paquete con flag SYN, la otra máquina acepta y envía SYN+ACK; entonces se establece una conexión. Esto se controla en IPTables mediante nuevos comandos:

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

Seguridad en la Capa de Acceso a Red: redes inalámbricas

Existe una amplia variedad de tecnologías y tipos de redes inalámbricas, entre las que destacan: Wi-Fi, Bluetooth, WiMAX, Zigbee… 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; además hay amenazas 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**. A grandes rasgos, el entorno inalámbrico tiene tres puntos de ataques:

- Cliente (estación)
- Punto de acceso (AP)
- Medio de transmisión

Las posibles amenazas son: robo de identidad, *Man-in-the-middle*, denegación de servicio, inyección en la red…

En cuanto a **medidas de seguridad** estas se pueden clasificar de acuerdo a las características de:

- La transmisión inalámbrica
- El punto de acceso (AP)
- Los elementos de interconexión (routers).

Medidas relacionadas con **la transmisión inalámbrica**:

- Amenazas:
    - La copia de mensajes.
    - La alteración.
    - La inserción de mensajes.
    - La interrupción de los mismos.
- Las contramedidas son:
    - Técnicas de ocultación de la señal.
    - Cifrado para el caso de las copias de mensajes.
    - Cifrado y autenticación para los casos de alteración o inserción.
    - Métodos contra DoS en el caso de interrupción.

Medidas relacionadas con el **AP:**

- La mayor amenaza es el **acceso no autorizado**.
- La forma de evitarlo es usando mecanismos de **autenticación** para los dispositivos que quieren conectar.

Medidas relacionadas con los **elementos de la red**:

- **Cifrado**: integrado en los routers inalámbricos, para el tráfico entre routers.
- **Deshabilitar el broadcast de identificación**: solo los dispositivos autorizados podrán conocer la identidad de los routers.
- **Dejar que solo los equipos específicos se conecten a la red**: solo direcciones cuyas MAC sean conocidas.
- **Cambiar el identificador**: sobre todo el *default* que trae el router de fábrica.
- **Cambiar la contraseña**: sobre todo la *default* que trae preestablecido.

IEEE 802.11

IEEE 802 es un comité que ha desarrollado estándares para diferentes tipos de redes LAN y WAN. 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.

Tipos de redes:

- **WWAN** 802.20 3G, 4G.
- **WMAN** 802.16 d,e,h,m WiMax.
- **WLAN** 802.11 a,b,gn WiFi.
- **WPAN** 802.15.1 Bluetooth.

Capas:

- *Logical Link Control*: control de flujo, control de errores
- *Medium Access Control*: ensamblar datos en *frame*, direccionamiento, detección de errores, acceso al medio.
- *Physical*: codificación/decodificación de señales, transmisión/recepción de bits, transmisión en el medio.

Diferencias red cableada y red inalámbrica:

	Cableada	**Inalámbrica**
Ventajas	● Robustez ● Ancho de banda ● Equipos baratos ● Fiables al contexto (ruido, vibración, interferencias, obstáculos)	● Rapidez y bajo coste de instalación ● Bajo coste de mantenimiento ● 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 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 e inalámbricas:

- En una **LAN cableada** hay una **intrínseca de autenticación** de las estaciones porque están directamente conectadas 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**; 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 a los de las LAN cableadas, limitado a la comunicación entre las estaciones 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 (en flujo) y tiene varias **vulnerabilidades**:

- Uso de *IV* débiles, que posibilitan que, a partir de un número de paquetes cifrados, el atacante pueda recuperar la clave secreta.
- Reutilización de *IVs*, debido a la corta longitud del *IV* y su concatenación con la clave secreta.

WPA (*Wi-FI Protected Access*)

Ante esa y otras debilidades iniciales, IEEE introdujo una serie de mejoras:

- Inicialmente, uso de **TKIP** (*Temporal Key Integrity Protocol*)
    - Se basa en la utilización de RC4, pero genera nuevas claves de cifrado con cierta periodicidad, por lo que elimina el problema de los *IV* débiles.
    - Además usa *IV* de 48 bits (en lugar de 24).
- Uso de AES como algoritmo de cifrado.
- Adopción del protocolo de autenticación 802.1X (desarrollado inicialmente para LAN cableadas).

Todas estas mejoras quedan recogidas en la versión IEEE 802.11i, en el protocolo **WPA.**

Servicios

El 802.11i define los 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**: influye el uso de la función de autenticación, el enrutado apropiado y facilita el intercambio de clave. Puede trabajar con distintos protocolos de autenticación.
- **Confidencialidad con integridad del mensaje**: los datos se cifran con una función MAC que asegura que los datos no han sido alterados.

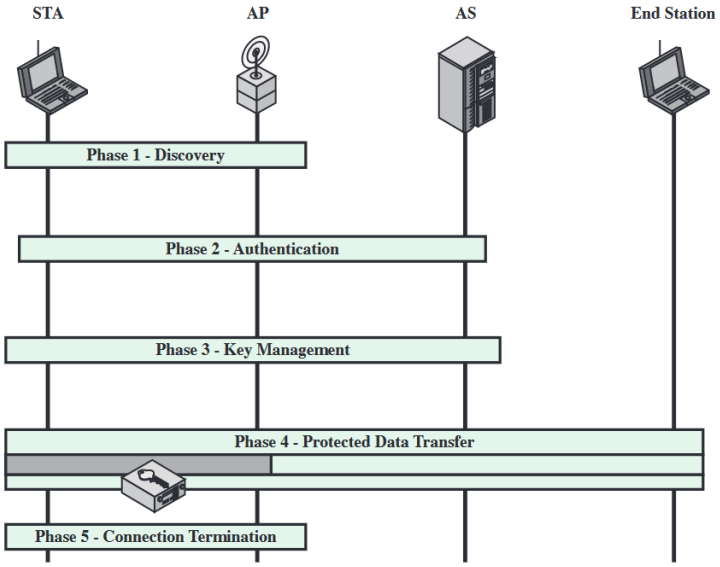
Funcionamiento

Las operaciones de seguridad en el 802.11i se distribuyen en 5 fases:

1. **Descubrimiento**.
    a. El AP utiliza mensajes llamados *beacons* y *probe responses* para anunciar su política de seguridad.
    b. La estación los utiliza para identificar al AP y se asocia con él seleccionando un *cipher suite* y un mecanismo de autenticación.
2. **Autenticación**
    a. La estación y el servidor de autenticación (AS) verifican la identidad del otro.
    b. 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**

a. 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**
    a. La estación origen y la estación final intercambian datos a través del AP pero la transferencia solo se realiza de forma segura (cifrada) entre la estación de origen y el AP.
5. **Finalización de la conexión**



La fase de **autenticación** presenta algunos detalles:

- 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 solo a estaciones autorizadas el uso de la red y garantizarles que están comunicando con una red legítima.
- El protocolo de autenticación es EAP (*Extensible Authentication Protocol*), definido en 802.1X

Este estándar define los términos *suplicant, authenticator* y *authentication server* para nombrar a la estación, AP y AS. El AS puede ser un dispositivo por sí mismo o ejecutarse dentro del AP.