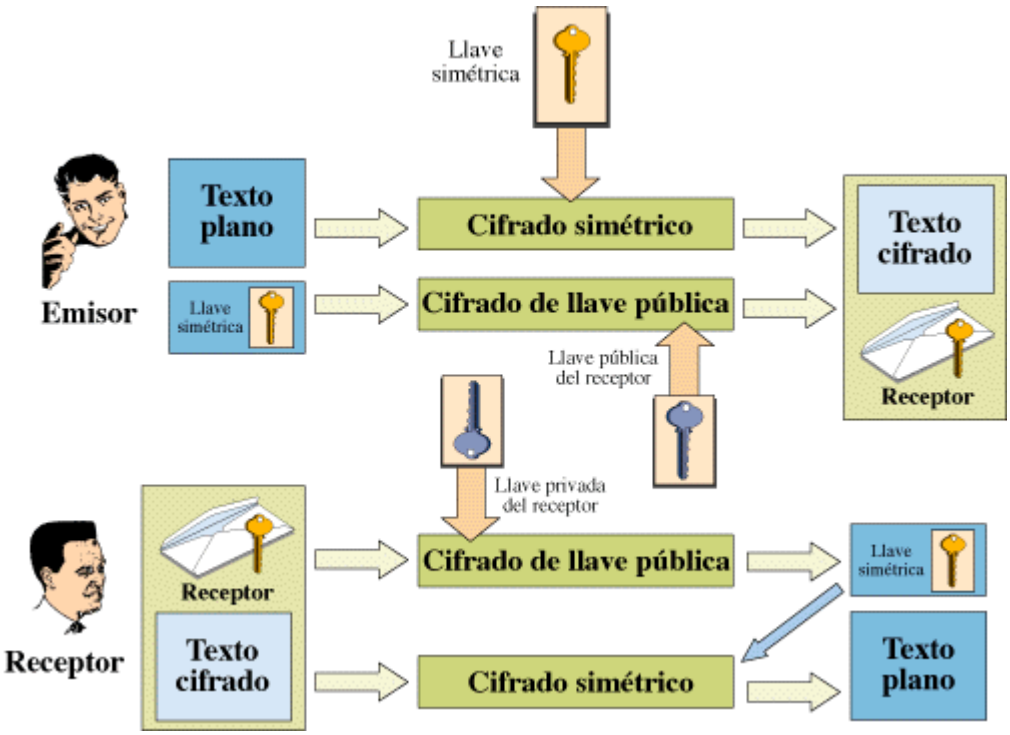
C A P Í T U L O VI

PROTOCOLOS SEGUROS



6.1 SSL *(Secure Sockets Layer)*

6.2 TLS *(Transport Layer Security)*

6.3 PCT *(Private Communications Technology)*

6.4 S-HTTP *(Secure HyperText Transfer Protocol)*

6.5 IPSEC *(IP Security)*

6.6 Conclusión de los Protocolos

6.1 SSL (Secure Sockets Layer)

SSL (Capa de Conexiones Seguras) es un protocolo de propósito general para establecer comunicaciones seguras, es decir, enviar información encriptada por Internet. Propuesto en 1994 por Netscape Communications Corporation [LIB 009]. SSL opera como una capa adicional entre el protocolo TCP/IP nativo y la capa de aplicación, tal como se muestra en la *figura 6.1*.

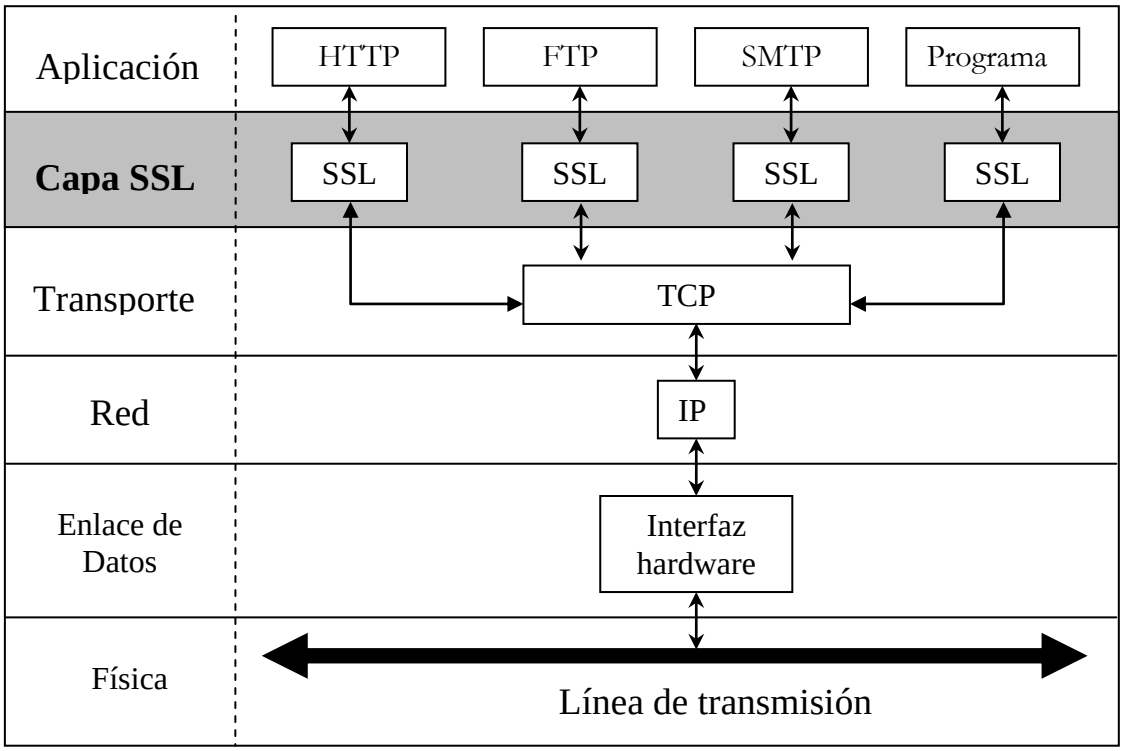


Figura 6.1 Capa SSL [LIB 009]

SSL proporciona las siguientes características al flujo de información:

- Autenticación y no repudiación del cliente y del servidor mediante firmas y certificados digitales.
- Privacidad de la transmisión mediante codificación de los datos.
- Integridad de los datos entre ambos extremos de una conexión.

SSL requiere un protocolo confiable de transporte (como TCP) para la transmisión y recepción de los datos; no puede correr sobre un protocolo no confiable, como UDP. El protocolo SSL se compone de dos capas. La capa de más bajo nivel, el *SSL Record Protocol*, que se encarga de encapsular los protocolos de nivel mas alto. La segunda capa, el *SSL Handshake Protocol*, que se encarga de la negociación de los algoritmos de encriptación, así como la autentificación entre el cliente y el servidor [WWW 023].

SSL HANDSHAKE PROTOCOL

Durante el *Handshake* se cumplen varios propósitos. Se hace autenticación del servidor y opcionalmente autenticación del cliente. Se determina que algoritmos de criptografía serán utilizados y se genera una llave secreta para ser utilizada durante el intercambio de mensajes subsiguientes durante la comunicación SSL. Los pasos que se siguen son los siguientes:

- **Cliente Hello**: El "saludo del cliente" tiene por objetivo informar al servidor que algoritmos de criptografía pueden utilizar y solicita una verificación de la identidad del servidor *(ver figura 6.2)*.

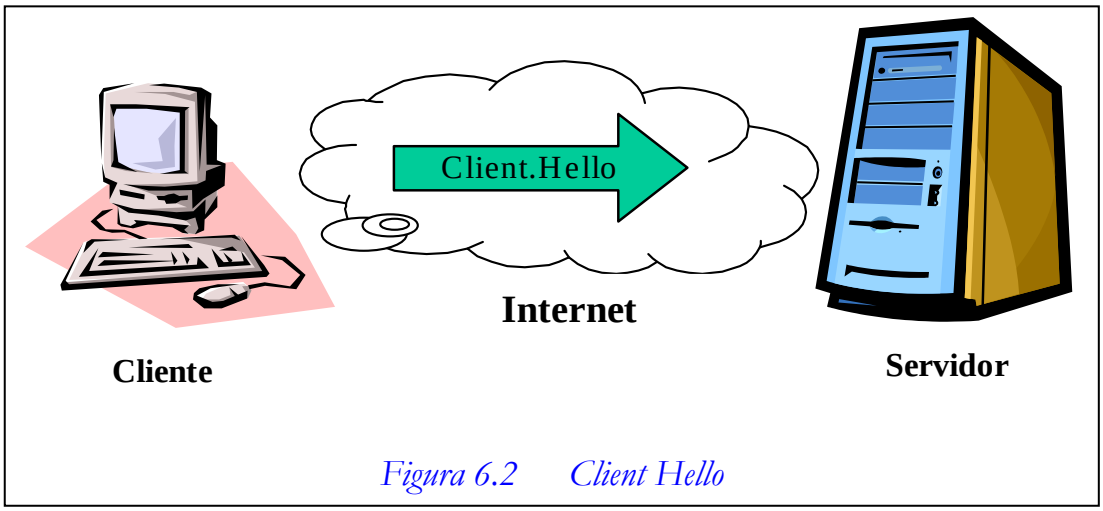


Figura 6.2 Client Hello

- **Servidor Hello:** El servidor responde enviando su identificador digital el cual incluye su llave pública, el conjunto de algoritmos criptográficos a usar. La decisión de que algoritmos serán utilizados está basada en el más fuerte que tanto el cliente como el servidor soporten. En algunas situaciones el servidor también puede solicitar al cliente que se identifique solicitando un identificador digital *(ver figura 6.3)*.

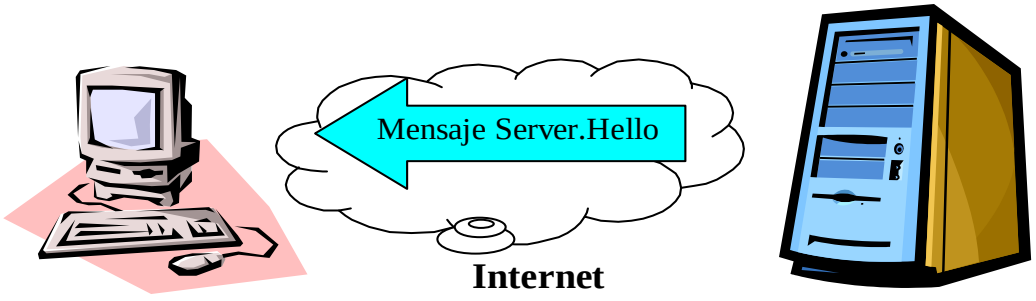


Figura 6.3 Server Hello

- **Aprobación del Cliente:** El cliente verifica la validez del identificador digital o certificado enviado por el servidor. Esto se lleva a cabo desencriptando el certificado utilizando la llave pública del emisor y determinando si este proviene de una entidad certificadora de confianza. Una vez que se ha verificado la autenticidad de la identidad del servidor, el cliente genera una llave aleatoria y la encripta utilizando la llave pública del servidor y el algoritmo criptográfico seleccionado anteriormente. Esta llave se le envía al servidor y en caso de que el Handshake tenga éxito será utilizada en el envío de futuros mensajes durante la sesión *(ver figura 6.4)*.

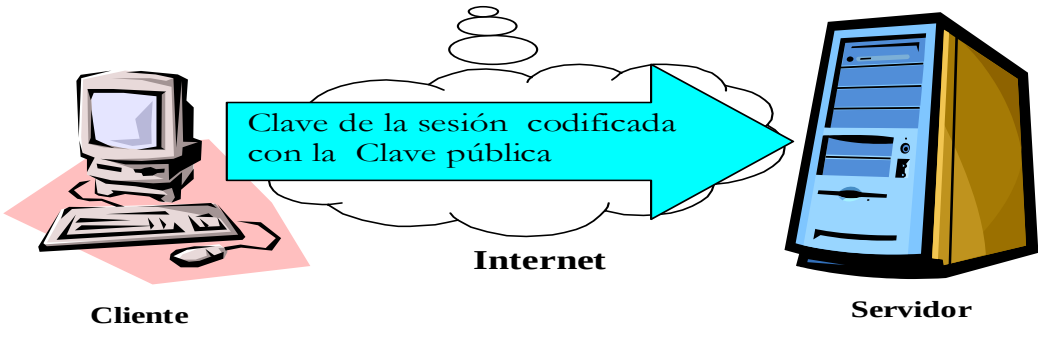


Figura 6.4. Aprobación del Cliente

- **Verificación:** En este punto ambas partes conocen la llave secreta, el cliente por que la generó y el servidor por que le fue enviada utilizando su llave pública. Se hace una última verificación para comprobar si la información transmitida hasta el momento no ha sido alterada. Ambas partes se envían una copia de las anteriores transacciones encriptada con la llave secreta. Si ambas partes confirman la validez de las transacciones, el Handshake se completa, de otra forma se reinicia de nuevo el proceso (*ver figura 6.5*).

Ahora ambas partes están listas para intercambiar información de manera segura utilizando la llave secreta acordada. El Handshake se realiza solo una vez y se utiliza una llave secreta por sesión [WWW 024].

SSL RECORD PROTOCOL

Ahora que se ha establecido un canal de transmisión seguro SSL, es posible el intercambio de datos. Cuando el servidor o el cliente desea enviar un mensaje al otro, se genera un Digest[25], se encripta el mensaje y el Digest y se envían. Cada mensaje es verificado utilizando el Digest.

[25] Resumen del mensaje para comprobar la integridad, utilizando un algoritmo de hash de una vía acordado durante el Handshake

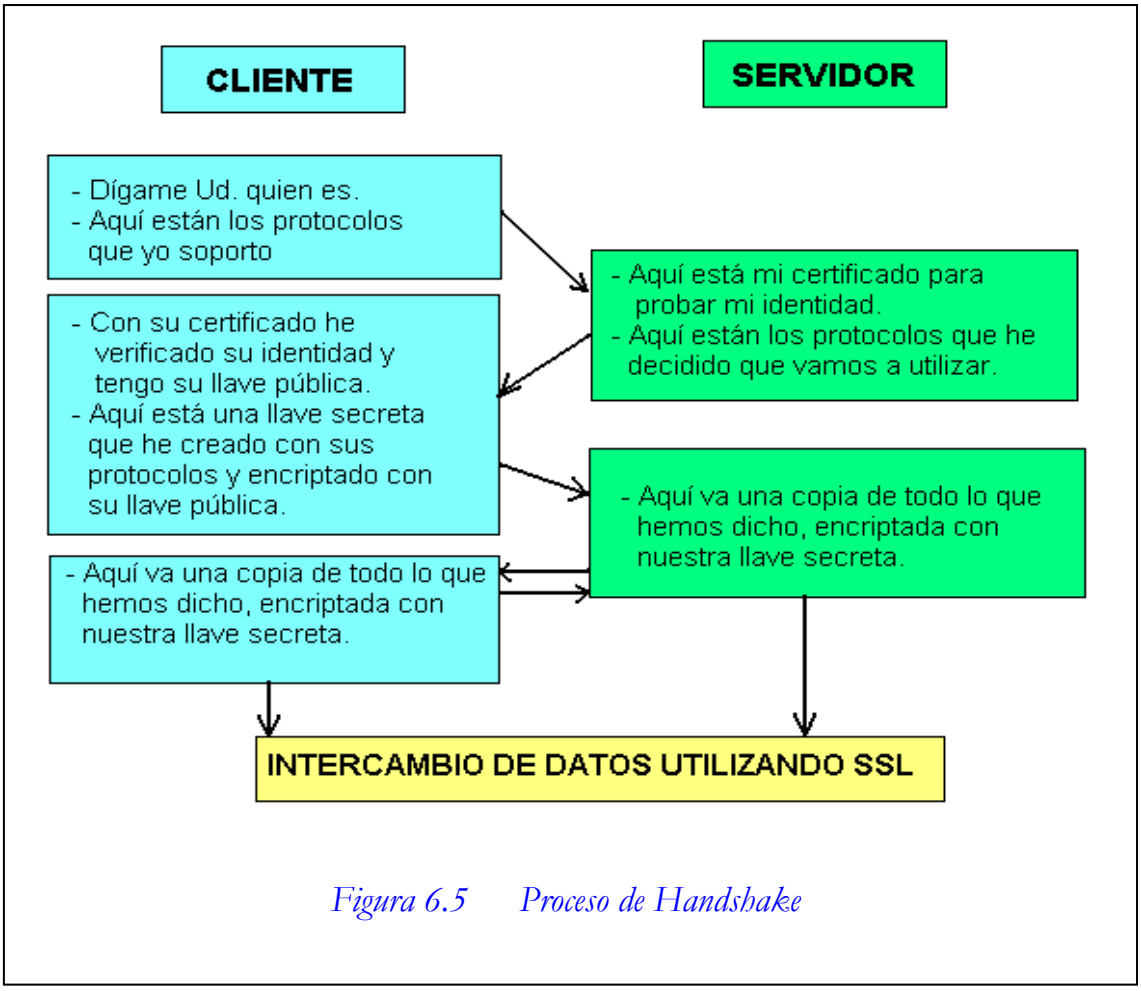


Figura 6.5 Proceso de Handshake

Si los usuarios utilizan un navegador de Internet que reconozca SSL como Netscape Navigator o Internet Explorer, pueden indicarle que cree una conexión encriptada con el servidor con solo remplazar el **"http"** del URL por **"https"**. Por ejemplo, si tenemos un documento localizado en la URL: http://www.utn.edu.ec/documento.html.
Los usuarios pueden obtenerlo en forma segura tecleando el URL: https:// www.utn.edu.ec/documento.html

Así como existe el servicio http protegido por SSL (https) existen otros servicios protegidos por SSL para los siguientes protocolos [WWW 022]:

Protocolo	Header	Puerto	Propósito
http	https	443/tcp	HTTP protegido por SSL.
smtp	ssmtp	465/tcp	SMTP protegido por SSL.
nntp	snntp	563/tcp	Grupos de noticias protegido por SSL
ldap	ssl-ldap	636/tcp	LDAP protegido por SSL.
pop3	spop3	995/tcp	POP3 protegido por SSL.

6.2 TLS (Transport Layer Security)

La última extensión de SSL es conocida con el nombre de Seguridad en la Capa de Transporte (TLS) [WWW 001]. El protocolo TLS basa su estructura en la especificación del Protocolo SSL 3.0, publicada por Netscape. Las diferencias entre ambos protocolos no son significativas, permitiendo que TLS y SSL sean interoperables. SSL se envió como un borrador al Cuerpo de Ingenieros de Internet (Internet Engineering Task Force IETF) y se convirtió en el estándar RFC 2246 (Transport Layer Security TLS 1.0) [WWW 024]. Ambos protocolos se diseñaron para proporcionar integridad y privacidad de datos entre dos aplicaciones que se comunican.

El Protocolo TLS se compone de dos capas: el Record Protocol de TLS 1.0, y el HandShake Protocol de TLS 1.0. Mientras el Record Protocol, se dedica a codificar y decodificar los mensajes que se trasmiten y reciben, el HandShake Protocol consiste de un conjunto de subprotocolos que se usan para permitir a las partes de la comunicación estar de acuerdo respecto a los parámetros de seguridad, autenticación y condiciones de informes de error. A continuación se detallan cada una de las capas de TLS.

EL RECORD PROTOCOL

Se encuentra a un nivel más bajo sobre un protocolo de transporte fiable (por ejemplo: TCP), proporcionando una conexión segura y privada. Usa criptografía simétrica para la encriptación de los datos (como: DES, RC4, etc.). El transporte del mensaje usa un mensaje de chequeo de integridad usando funciones de hash seguras como SHA y MD5. Este protocolo se usa para la encapsulación de varios protocolos de más alto nivel: HandShake Protocol y el protocolo de la aplicación. Fundamentalmente se encarga de tomar los mensajes a ser transmitidos, fragmenta los datos en bloques manejables, opcionalmente comprime los datos, encripta y transmite el resultado. El dato recibido es desencriptado, verificado, descomprimido y reensamblado, entregándose a los clientes en niveles más altos.

EL HANDSHAKE PROTOCOL

Este protocolo involucra el uso del Record Protocol de TLS para intercambiar una serie de mensajes entre un servidor y un cliente al comienzo de una conexión TLS. Este intercambio de mensajes se diseñó para permitir las siguientes acciones:

- La autenticación del servidor por parte del cliente.
- Permitir que el cliente y el servidor seleccionen los algoritmos criptográficos, que ambos puedan soportar.
- La autenticación opcional del cliente por parte del servidor.
- Usar técnicas de encriptación de clave pública para generar secretos compartidos.
- Establecer una conexión TLS encriptada.

DIFERENCIAS ENTRE SSL Y TLS [WWW 025]

	SSL v3	TLS v1
Errores de Alerta	Si el servidor ha enviado un mensaje de requerimiento de certificado, el cliente debe enviar en su lugar el mensaje certificado o un alerta no-certificate[26].	Una vez que el servidor ha enviado un mensaje de pedido de certificado el cliente deberá enviar el mensaje certificado
Algoritmos de Intercambio de Claves	En SSL se tienen como algoritmos de intercambio de claves RSA, Diffie-Hellman y Fortezza[27].	TLS no soporta el algoritmo de intercambio de claves Fortezza.
El MAC [28] se calcula diferente	En el cálculo del MAC no se incluye algunos campos, por lo tanto estos campos no están protegidos contra ataques a la integridad.	Se protegen todos los campos, incluyéndolos en el cálculo del MAC.

Como hemos visto, el protocolo TLS 1.0 se basa en el SSL 3.0, el cual provee excelente seguridad. Los cambios presentados por el protocolo TLS respecto de su antecesor refuerzan la seguridad e integridad. Adicionalmente, se ha tomado cuidado de reducir la actividad de la red por medio de un esquema de cache de sesiones opcionales.

Actualmente tanto SSL como TLS son utilizados, entre otras cosas, en la realización de transacciones comerciales con tarjetas de crédito, soportando solo el transporte de los datos entre el cliente y el servidor. Existe una

[26] Un mensaje de alerta si no es apropiado el certificado disponible.
[27] El Fortezza es similar al Diffie-Hellman, con valores públicos fijos, contenidos en los certificados.
[28] Message Authentication Code (Código de Autenticación de Mensaje).- mensaje de chequeo de integridad, usando funciones de hash seguras como SHA y MD5

creciente necesidad de que dichos datos se mantengan seguros aún después de ser entregados (por ejemplo, en el caso de compras por Internet con tarjetas de crédito). Para cubrir esta necesidad, protocolos como SET (Secure Electronic Transaction) van a popularizarse en el futuro para este tipo de transacciones, dejando a TLS y SSL sólo dedicados a la seguridad en el transporte de datos [WWW 025].

6.3 PCT (Private Communications Technology)

Microsoft Corporation creó el protocolo Tecnología de Comunicación Privada (PCT) para evitar las escuchas electrónicas en aplicaciones cliente/servidor. Este protocolo es compatible con SSL, pero exige corregir o mejorar varios puntos débiles de SSL.

La finalidad de PCT es proporcionar una vía de acceso de comunicación privada entre un cliente y un servidor. El protocolo incorpora autentificación al servidor y ofrece la misma opción también a los clientes. Al igual que SSL, PCT requiere un protocolo de transmisión fiable, como TCP, y ambos son independientes del protocolo de aplicación, por lo que protocolos de aplicaciones de nivel más alto como HTTP o FTP pueden superponerse y operar de forma transparente.

PCT inicia las conexiones estableciendo una comunicación para negociar algoritmo y clave de encriptación simétrica y luego autentificando el servidor. La diferencia entre PCT y SSL, en este aspecto radica en que, al establecer esta comunicación, PCT emplea claves públicas asimétricas certificadas. Esta medida ayuda a PCT a resolver uno de los problemas de seguridad de SSL. El protocolo PCT no especifica detalles en relación con la verificación del certificado. PCT espera que el programador aporte una función que decida la validez de los certificados recibidos. Aplicar sus propias normas de validación

en realidad es una ventaja, porque le ofrece la opción de elegir un sistema de certificación en función de sus propias necesidades. Tras el establecimiento de la comunicación, PCT encripta todas las transmisiones de datos empleando la clave de sesión negociada durante el proceso de establecimiento de la comunicación.

PCT se diferencia de SSL, principalmente en la fase del establecimiento de la comunicación. Fuera de este aspecto, la seguridad es bastante parecida. *Las principales diferencias entre PCT y SSL son* [WWW 023]*:*

- La estructura del mensaje de PCT es apreciablemente más corta que la de SSL.

- PCT ofrece una gama más amplia en los algoritmos y los formatos criptográficos negociados. Además del tipo de cifrado y del tipo de certificado del servidor que se negocian en las sesiones de SSL, PCT negocia el tipo de función hash y el tipo de intercambio de claves. Si se necesita la autentificación de cliente, PCT también negocia el tipo de firma de cliente y el tipo de certificado.

- Para la autentificación de mensajes se emplean claves distintas de las de encriptación. Esto posibilita claves de autentificación de mayor longitud, lo que hace que el proceso de autentificación sea mucho más seguro.

- En la secuencia interrogación/respuesta de la autentificación de cliente de PCT se emplea el tipo de cifrado que se ha negociado para la sesión. En la autentificación de cliente de SSL se emplea un cifrado más débil que es independiente del tipo que se haya negociado para la sesión.

6.4 S-HTTP (Secure HyperText Transfer Protocol)

El Protocolo de Transferencia de HiperTexto Seguro (S-HTTP) es una capa de extensión sobre el estándar HTTP para la transferencia Web, con el propósito de encriptar la información durante una sesión HTTP. Al igual que SSL, permite tanto el cifrado como la autentificación digital. Sin embargo, a diferencia de SSL, S-HTTP es un protocolo de nivel de aplicación, es decir, que extiende el protocolo HTTP por debajo, incorporando cabeceras MIME[29] para aportar confidencialidad, autenticación, integridad e irrenunciabilidad de las transacciones.

S-HTTP soporta una gran variedad de mecanismos de seguridad para los clientes y servidores de HTTP, ofreciendo las opciones de servicios de seguridad apropiados al amplio rango de usos finales potenciales del WWW. El protocolo ofrece capacidades simétricas tanto para el cliente como para el servidor. Es totalmente compatible con HTTP, por lo que un cliente que utilice S-HTTP puede acceder a un servidor que no lo utilice y viceversa, es decir, un servidor que utilice S-HTTP puede servir a clientes que no lo utilicen.

No se requiere el uso de cifrado asimétrico, por lo cual no es necesario la utilización de claves públicas, y es fácil realizar transacciones privadas espontáneas entre usuarios, sin necesidad de que exista una clave pública establecida. S-HTTP soporta transacciones seguras extremo a extremo. Ofrece una gran flexibilidad a la hora de utilizar algoritmos, modos y parámetros criptográficos; el servidor y el cliente deben negociar dichas características.

[29] Secure / Multipurpose Internet Mail Extension (Extensiones Seguras Multipropósito de Correo en Internet)

Los mensajes se pueden proteger mediante firma digital, autenticación y cifrado, o cualquier combinación de las tres. Se incluyen mecanismos para el control de claves múltiples, que incluyen palabras clave secretas compartidas manualmente, mecanismos de intercambio de claves por clave pública y el sistema Kerberos. El protocolo también permite controlar el tiempo de validez de los mensajes, mediante mecanismos de preguntas-respuestas e introducción de la fecha-hora en la cabecera.

Servidor y cliente pueden ponerse de acuerdo en cuanto a los requerimientos y preferencias en los elementos criptográficos que se van a utilizar. Sin entrar en los mecanismos y los pasos de la negociación, vamos a describir brevemente algunos de los parámetros que se pueden negociar:

- Tipos de certificados de clave pública que se aceptarán. Actualmente, el único tipo permitido es X.509.

- Tipos de algoritmos que se podrán utilizar para el intercambio de claves. Actualmente, se permiten "RSA", "Outband", "Inband" y "Krb" (Kerberos).

- Tipos de algoritmos de firma digital. Los valores permitidos son "RSA", y "NIST-DSS".

- Tipos de algoritmos de cálculo de un código hash (resumen). Los valores permitidos son "RSA-MD2", "RSA-MD5" y "NIST-SHS".

- Tipos de algoritmos de cifrado simétrico para cifrar el mensaje. Los valores definidos son "DES-CBC", "IDEA-CFB", "RC4", etc.

Las principales ventajas de S-HTTP son su flexibilidad y su integración dentro de HTML (extensiones al lenguaje similares a las introducidas periódicamente por Netscape en sus navegadores). Entre sus debilidades podemos señalar los efectos derivados de mantener la compatibilidad hacia atrás y la necesidad de

implementar servidores que soporten las extensiones a HTML aportadas por el protocolo S-HTTP [WWW 003].

6.5 IPSEC (IP Security)

IPSec es un grupo de extensiones de la familia del protocolo IP. IPSec provee servicios criptográficos de seguridad similares a SSL, pero a nivel de redes, de un modo que es completamente transparente para las aplicaciones y mucho más robusto (*ver figura 6.6*). Puede crear túneles cifrados (VPNs)[30], o simple cifrado entre computadoras [WWW 002].

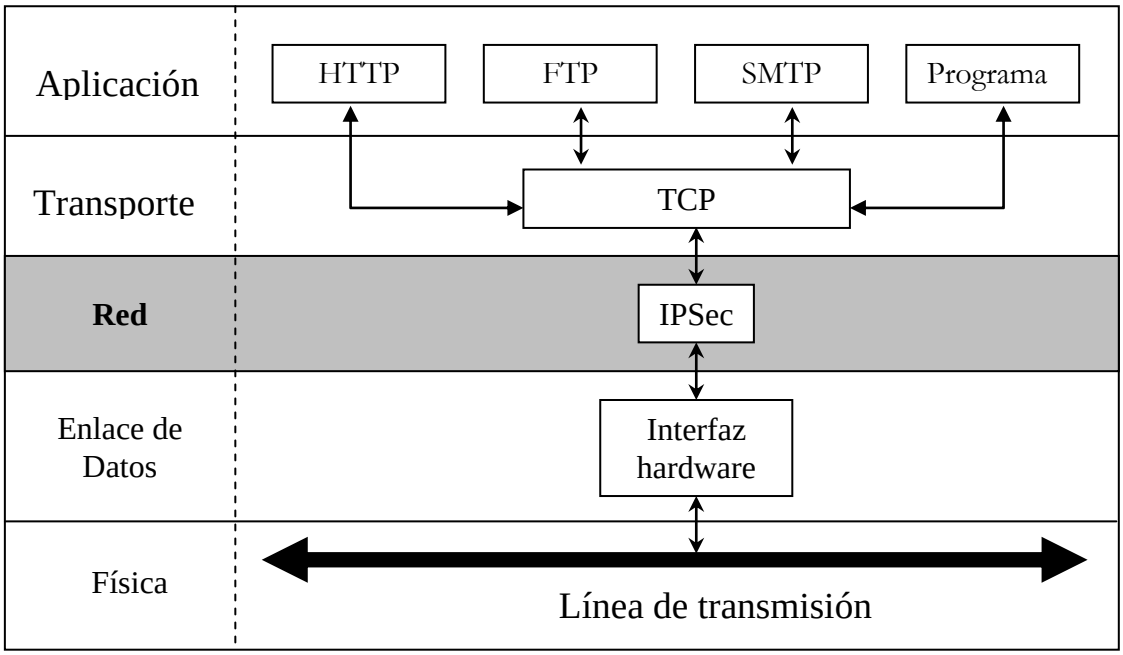


Figura 6.6 Capa IPSec [LIB 009]

IPSec proporciona los siguientes servicios:

- Confidencialidad de la transmisión mediante codificación para los datos.
- Integridad.- Garantiza que los datos no puedan ser cambiados en el viaje.

[30] Redes virtuales privadas.

- Autenticidad.- Firma sus datos de modo que otros puedan verificar quien lo envía.
- Protección a la réplica.- Asegura que una transacción sólo se puede llevar a cabo una vez. Nadie debería poder grabar una transacción, y luego replicarla al pie de la letra, con el propósito que pareciera como si se hubieran recibido múltiples transacciones del remitente original.

IPSec provee confidencialidad, integridad, autenticidad, y protección a la réplica a través de dos nuevos protocolos (*ver figura 6.7*). Estos protocolos se llaman: Cabecera de Autenticación (AH, "Authentication Header") definido en la RFC1826 y Cargo de Seguridad Encapsulado (ESP, "Encapsulated Security Payload") definido en la RFC1828 [WWW 002].

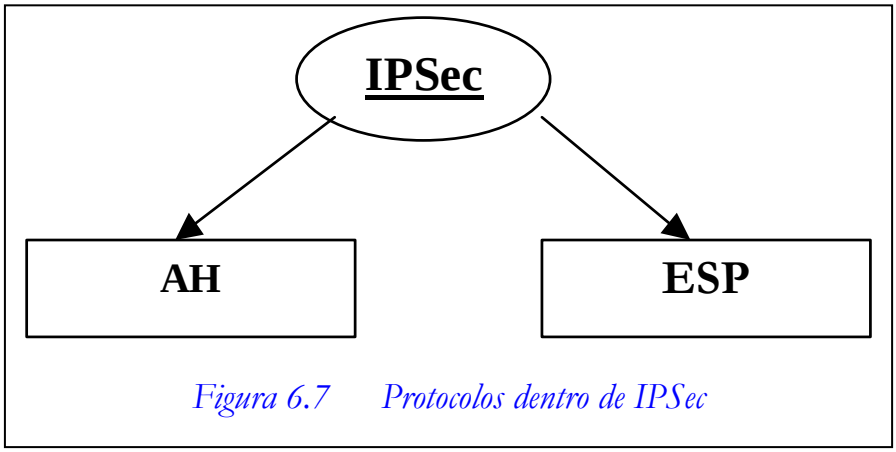


Figura 6.7 Protocolos dentro de IPSec

Cabecera de Autenticación (AH) provee autenticación, integridad, y protección a la réplica (pero no confidencialidad). Contiene resúmenes (hash) de los datos e información de identificación de la trama[31].

Cargo de Seguridad Encapsulado (ESP) puede proveer autenticación, integridad, protección a la réplica, y confidencialidad de los datos (asegura

[31] Unidad de transferencia que contiene en su cabecera direcciones físicas (direcciones MAC). Encapsula al datagrama IP y se encuentra en la Capa de Enlace.

todo lo que sigue a la cabecera en el paquete). La cabecera del ESP permite reescribir el cargo en modo cifrado. La cabecera ESP no considera los campos de la cabecera IP que van delante, y por lo tanto no garantiza nada excepto el cargo.

Tanto el protocolo ESP como el protocolo AH pueden ser usados en dos modos: Modo Túnel y Modo Transporte.

En el **Modo Túnel**, el paquete IP entero es encriptado y entonces situado dentro de otro paquete IP. Mirándolo de otro modo, el paquete entero IP es encriptado y una nueva cabecera IP es añadida, es decir, todo el datagrama[32] original es cifrado y encapsulado en otro datagrama con cabeceras no cifradas. La dirección de destino de Internet en la añadida cabecera IP debe ser un host, pero puede ser una entrada de seguridad (security gateway), tal como un cortafuegos o un router habilitado. De esta manera, incluso si un paquete es capturado, el interceptor no leerá mucho por examinación de la dirección de destino. El interceptor no será capaz de decir cual host detrás de la entrada de seguridad (security gateway) recibirá el paquete IP.

En el **Modo de Transporte**, el paquete IP es enviado directamente al destino sin ser encapsulado. La cabecera principal IP es seguida por el ESP o la cabecera AH y entonces por el resto del paquete IP (encriptado en el caso de ESP).

[32] Unidad básica de transferencia en una red de redes TCP/IP, se divide en áreas de encabezado y datos. Se encuentra en la Capa de Red. El encabezado contiene direcciones IP.

ASOCIACIONES DE SEGURIDAD (SA)

- Una SA la compone la información que necesita una entidad IPsec para soportar un sentido del tráfico (saliente o entrante) de una conexión de un protocolo IPsec.
- El contenido de una SA variará para cada conexión, y puede incluir claves de autenticación o cifrado, algoritmos específicos, tiempos de vida de las claves, direcciones IP.
- Una SA indica a un dispositivo IPsec cómo procesar paquetes IPsec entrantes, o cómo generar paquetes IPsec salientes.
- Los dispositivos IPsec insertan un campo en la cabecera de IPsec (Índice de Parámetros de Seguridad) para asociar un cierto datagrama a la SA adecuada en las máquinas que los procesen.
- Los dispositivos IPsec almacenan las SAs en una base de datos (SAD).

6.5.1 CABECERA DE AUTENTICACIÓN (AH)

Su propósito es:

- Detectar alteraciones en el contenido de un paquete.
- Autenticar la identidad del que envía, sea como usuario o por su dirección IP.

Esta cabecera se coloca detrás de todas las opciones de la cabecera IP, como se muestra en la *figura 6.8*.

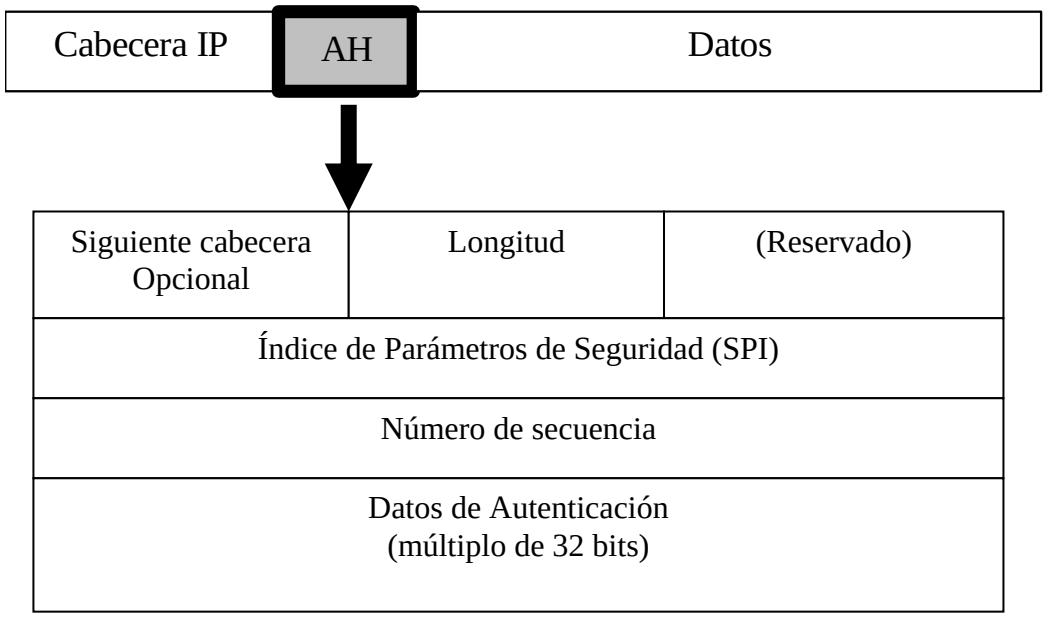


Figura 6.8 Componentes del Protocolo Cabecera de Autentificación (AH)

- **Longitud:** Tamaño de toda la AH en palabras de 32 bits, menos 2.
- **SPI:** Junto con la dirección IP destino y el protocolo AH, identifica de manera única una Asociación de Seguridad (SA) para este datagrama.
- **Número de secuencia:** Se inicializa a 0 al establecerse una SA. Nunca debe dar la vuelta. Se usa para evitar que alguien repita paquetes en la red (protección anti-repetición).
- **Datos de autenticación:** Firma hash.

Posición de la cabecera AH en Modo Transporte

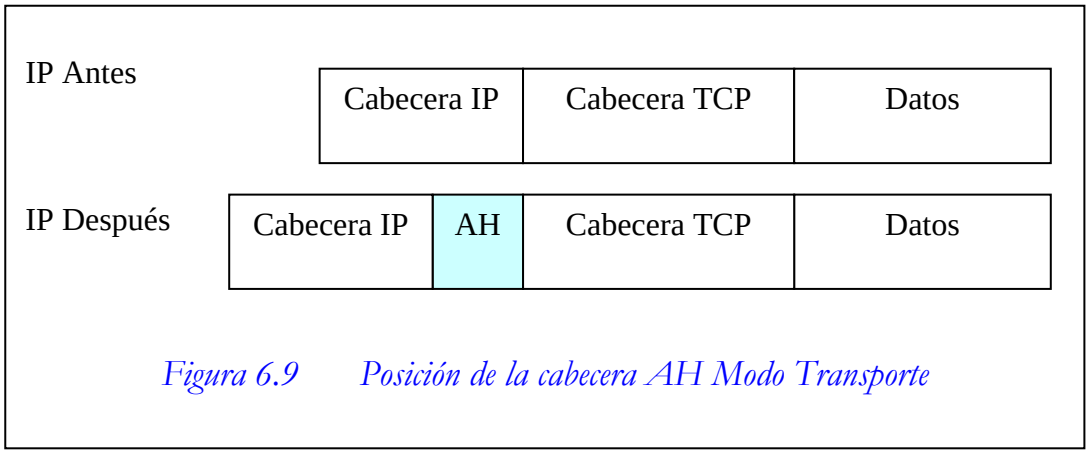


Figura 6.9 Posición de la cabecera AH Modo Transporte

Posición de la cabecera AH en Modo Túnel

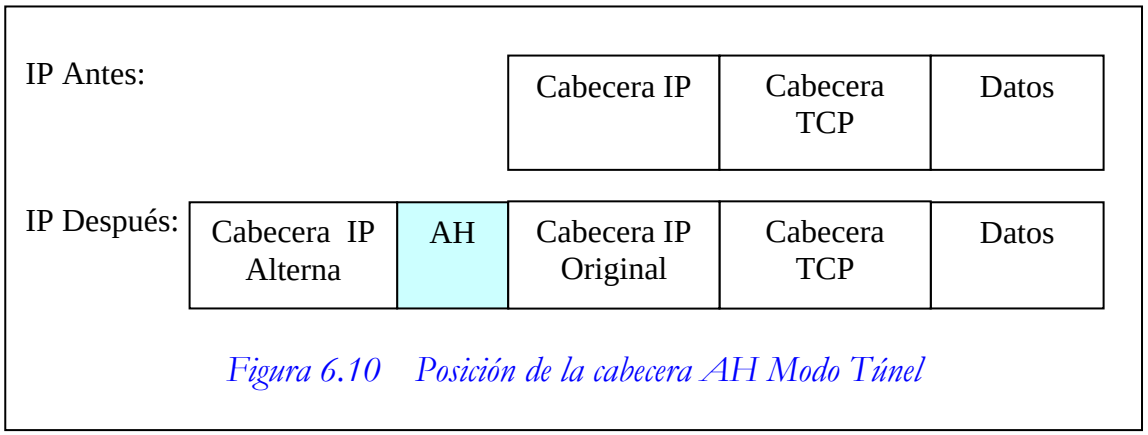


Figura 6.10 Posición de la cabecera AH Modo Túnel

Envío de un paquete con AH

Lo puede hacer el originador del paquete, o una pasarela (gateway) intermedia:

- Identificar el algoritmo (ejemplo: MD5 ó SHA-1) y la clave para la SA determinada por el SPI del datagrama.
- Incrementar el número de secuencia.
- Identificar la información a firmar.

- Insertar la cabecera AH en su lugar.

- En su caso, fragmentar después de tener la AH si se es el originador. En modo túnel es posible que se aplique el AH a un fragmento.

Recepción de un paquete con AH

Lo puede hacer el destinatario del paquete, o una pasarela (gateway) intermedia:

- En su caso, reensamblar antes de comprobar el AH. Si al ir a comprobarlo se ve que es sobre un fragmento, descartarlo.

- Comprobar el número de secuencia.

- Identificar la información firmada.

- Identificar el algoritmo (ejemplo: MD5 ó SHA-1) para la SA determinada por el SPI del datagrama, y la clave secreta.

- Calcular la firma, y si no coincide con la que viene en la AH, descartar el datagrama.

Comprobación del número de secuencia (protección anti-repetición)

Se establece una ventana:

- Límite superior: último número de secuencia válido recibido en la SA.

- Límite inferior: Al menos, 32 números de secuencia menos que el límite superior

Al recibir un cierto número de secuencia:

- Si el número es menor que el límite inferior de la ventana, se descarta el datagrama.
- Si el número cae dentro de la ventana y está repetido, se descarta el datagrama.
- Si el número cae dentro de la ventana y es nuevo, se comprueba la autenticación.
- Si el número es mayor que el límite superior de la ventana, se comprueba la autenticación, y si es correcta se actualizan los límites de la ventana.

6.5.2 ENCAPSULACIÓN DE LA CARGA DE SEGURIDAD (ESP)

Su propósito es proporcionar:

- confidencialidad
- autenticidad
- integridad
- confidencialidad del flujo del tráfico (en modo túnel)

El formato de ESP incluye todo el datagrama, excepto la cabecera IP, como se muestra en la *figura 6.11*.

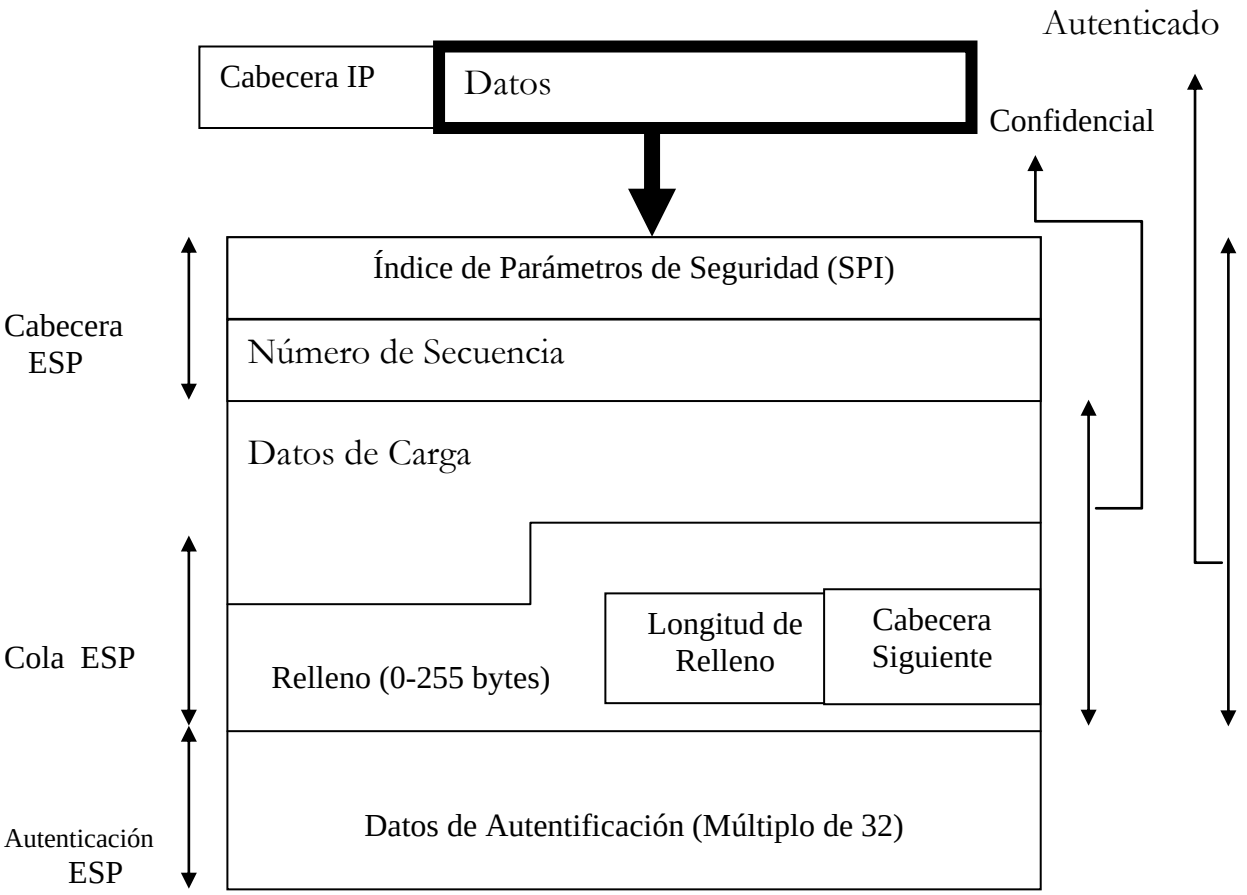


Figura 6.11 Componentes del Protocolo Encapsulación de la Carga de Seguridad (ESP)

- **SPI:** Junto con la dirección IP destino y el protocolo ESP, identifica de manera única una SA para este datagrama. Normalmente es elegido por el destinatario cuando se establece la SA.

- **Número de secuencia:** Se inicializa a 0 al establecerse una SA. Nunca debe dar la vuelta. Se usa para evitar que alguien repita paquetes en la red (anti-repetición).

- **Datos de carga útil:** Datos que se transmiten.

- **Cabecera siguiente:** Número de protocolo de nivel superior en los Datos de la carga útil.

- **Datos de autenticación:** Opcional, sólo si lo establece la SA. En este caso, no se precisa AH.

Posición de ESP en Modo Transporte

- Sólo para implementaciones en máquinas finales.
- No proporciona confidencialidad a la cabecera IP.

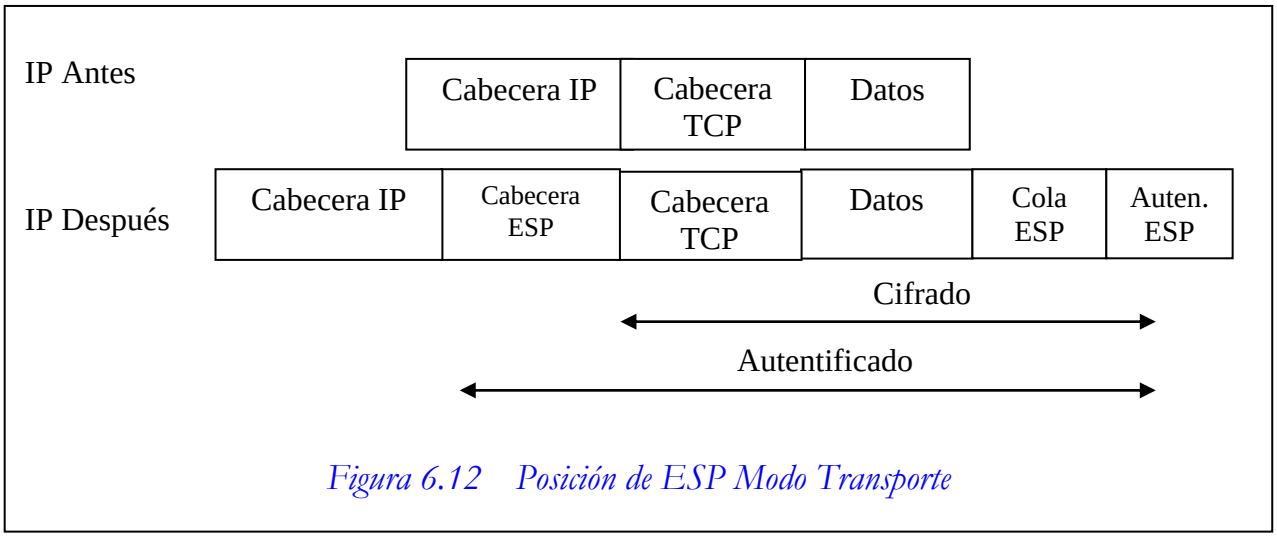


Figura 6.12 Posición de ESP Modo Transporte

Posición de ESP Modo Túnel

- Para implementaciones en máquinas finales o en pasarelas.
- Proporciona confidencialidad también a la cabecera IP.

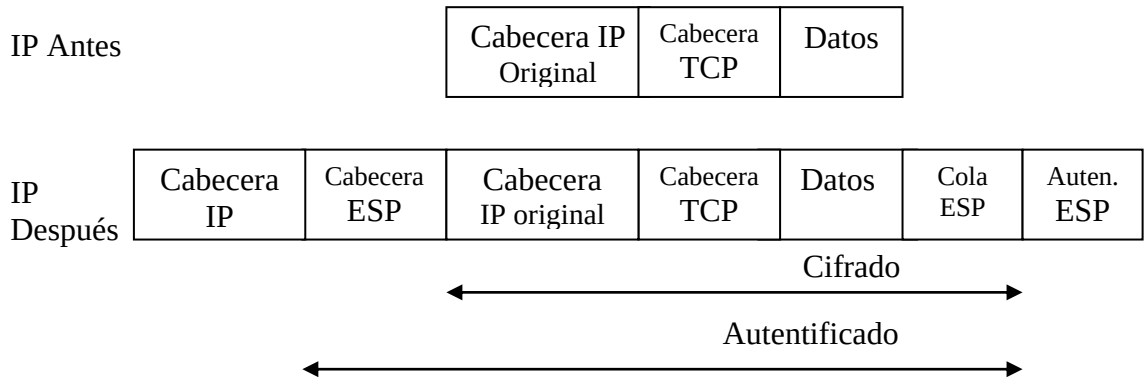


Figura 6.13 Posición de ESP Modo Túnel

Envío de un Paquete ESP

Lo puede hacer el originador del paquete, o una pasarela (gateway) intermedia:

- Identificar si se usa cifrado y, en su caso, el algoritmo (ejemplo: DES) y la clave para la SA determinada por el SPI del datagrama.
- Identificar si se usa autenticación y, en su caso, el algoritmo (ejemplo: MD5, SHA-1) y la clave para la SA determinada por el SPI del datagrama.
- Incrementar el número de secuencia.
- Preparar la carga útil, añadiendo el relleno en caso necesario (en modo túnel, la carga útil es todo el datagrama, y hay que preparar la nueva cabecera).
- En su caso, cifrar la carga útil.
- En su caso, calcular la firma de la cabecera ESP y la carga útil.
- Colocar la cabecera ESP, la carga útil, la cola ESP y la autenticación ESP en su lugar.

- En su caso, fragmentar después de aplicar ESP si se es el originador. En modo túnel es posible que se aplique ESP a un fragmento.

Recepción de un Paquete ESP

Lo puede hacer el destinatario del paquete o una pasarela (gateway) intermedia:

- En su caso, reensamblar antes de procesar ESP. Si al ir a procesarlo se ve que es sobre un fragmento, descartarlo.
- Comprobar el número de secuencia.
- En su caso, identificar el algoritmo de autenticación para la SA determinada por el SPI del datagrama, y la clave secreta, calcular la firma, y si no coincide descartar el datagrama.
- En su caso, identificar el algoritmo de cifrado y descifrar el datagrama; en caso de error descartar el datagrama.

6.6 Conclusión de los Protocolos

Protocolo	¿Qué es?	Algoritmos	¿Qué Proporciona?	Capa	Fabricante	RFC
SSL *(Secure Sockets Layer)*	Protocolo para encriptar transmisiones TCP/IP	RSA, Diffie-Hellman, RCZ, RC4, MD5	Confidencialidad, autenticación, integridad, no repudiación.	Aplicación	Netscape Communications Corporation	2104 2246
TLS *(Transport Layer Security)*	Protocolo para encriptar transmisiones TCP/IP	DES, RC4, SHA , MD5	Confidencialidad, autenticación, integridad, no repudiación.	Aplicación	Internet Engineering Task Force (IETF)	2246
PCT *(Private Communications Technology)*	Protocolo para encriptar transmisiones TCP/IP	RSA, MD5, RCZ, RC4	Confidencialidad, autenticación, integridad, no repudiación.	Aplicación	Microsoft	2104 2246 2630
S-HTTP *(Secure HyperText Transfer Protocol)*	Protocolo para encriptar peticiones y respuestas HTTP.	RSA, DES, IDEA	Confidencialidad, autenticación, integridad, no repudiación; Sin embargo es obsoleto.	Aplicación	Enterprise Integration Technologies (EIT)	2660 1866
IPSEC *(IP Security)*	Protocolo de bajo nivel para encriptar paquetes IP	Diffie-Hellman	Confidencialidad, autenticación, integridad, no repudiación.	Red	Internet Engineering Task Force (IETF)	2401