# Seguridad en redes de Internet

# Seguridad en redes de Internet

## Conceptos básicos sobre seguridad

## Servicios de seguridad

Se entiende por servicio de seguridad aquél que mejora la seguridad de los sistemas de procesado y de comunicación de la información de un grupo determinado de personas o de una organización. Un servicio de seguridad hace frente a ataques contra la seguridad empleando uno o varios mecanismos.
En el estándar de arquitectura de seguridad de la ISO se definen cinco servicios:

1- **Confidencialidad:** Protege contra el acceso no autorizado a parte o a la totalidad de la información.
2- **Autenticación:** Proporciona la certeza de la identidad de una entidad. Puede ser simple o mutua. Cuando es simple, únicamente uno de los comunicantes tiene que demostrar su identidad. Cuando es mutua, ambos comunicantes se identifican el uno al otro.
3- **Integridad:** Protege contra la modificación, el borrado o la sustitución de la información durante la transmisión. Igual que ocurre con el servicio de confidencialidad, se puede aplicar a una cadena de mensajes, a un único mensaje, o a campos específicos del mensaje.
4- **No repudio:** Protege contra la negación de una entidad que participa en una comunicación, de haber enviado un mensaje (repudio de origen) o de haberlo recibido (repudio de entrega). Dicho servicio también es conocido como irrenunciabilidad.
5- **Control de acceso:** Protege contra el uso o la manipulación no autorizada de recursos. Se controla el acceso a los sistemas informáticos y a los enlaces de datos evitando las infiltraciones y el uso no autorizado de recursos.

En la Tabla 5.1 se muestra una serie de analogías entre los servicios de seguridad y la vida cotidiana.

| Servicio de seguridad | Ejemplo de la vida cotidiana |
|---|---|
| Autentificación | Carné con identificación fotográfica<br>Huellas dactilares |
| Control de acceso | Llaves y cerrojos |
| Confidencialidad | Tinta invisible<br>Carta lacrada |
| Integridad | Tinta indeleble |

| | |
|---|---|
| No repudio | Firma notarizada<br>Correo certificado |

*Tabla 5.1. Analogías entre servicios de seguridad y vida cotidiana.*

La criptología es una ciencia constituida por dos subciencias antagonistas: la criptografía y el criptoanálisis. Las técnicas criptográficas constituyen uno de los mecanismos más utilizados para proveer dichos servicios. Por ello, serán presentadas con detalle posteriormente, aunque a continuación se darán unas breves pinceladas sobre estos conceptos.

**Criptografía:**

La ciencia de la criptografía estudia las comunicaciones electrónicas, principalmente digitales, que se realizan en un medio hostil, vulnerable y en el que existe una desconfianza entre las entidades comunicantes.

1- **Hostil**: pueden existir atacantes que quieran evitar que la comunicación tenga lugar.
2- **Vulnerable**: los atacantes pueden desear modificar la información transmitida u obtenerla de forma ilícita.
3- **Desconfianza**: un participante puede intentar perjudicar al otro.

Los sistemas analizados por la criptografía enmascaran la información con el objetivo de garantizar una serie de requisitos, como la confidencialidad, integridad, autenticación, etc.

**Criptoanálisis:**

El criptoanálisis trata de romper los sistemas que la criptografía implementa para así poder obtener, por ejemplo, la información enmascarada por tales técnicas.

## Ataques y amenazas

Tanto en la transmisión de datos como en el almacenamiento de éstos, se deben asegurar dos requisitos fundamentales: la confidencialidad y la autenticación, para prevenir el acceso a la información o su modificación de forma no autorizada.
Los peligros en la transmisión y almacenamiento de datos pueden ser debidos tanto a la actuación de terceras partes con fines no amistosos (atacantes), a fallos de los sistemas telemáticos o bien a la incorrecta utilización de éstos.
Los ataques de los que pueden ser objetos los sistemas telemáticos se pueden dividir en dos categorías:

- **Pasivos**: cuando el atacante o fisgón (*eavesdropper*) simplemente observa o escucha la información, pero no la manipula. En estos casos, no existen problemas de autenticidad.
- **Activos**: cuando el atacante realiza algún tipo de modificación sobre la información transmitida. Estos ataques pueden ser de tres tipos básicos:
  - Suplantación de identidad.
  - Manipulación de la información, ya sea reemplazando, eliminando, insertando o reordenando los datos.
  - Reactuación, o sea, la grabación de una comunicación con el objetivo de repetirla posteriormente (a esto hace referencia el servicio de idempotencia).

En lo que hace referencia al almacenamiento de datos, los riesgos más destacables son los siguientes:

- Búsqueda (*browsing*) de información confidencial por parte de los usuarios que tienen acceso al sistema pero no poseen la autorización necesaria para observar dicha información.
- Manipulación (*tampering*) de la información: modificación, inserción, reemplazo o eliminación de datos.
- Impostura o suplantación (*masquerading*): cuando un usuario se hace pasar por otro para obtener privilegios adicionales.

## Objetivos de la criptografía

Las tecnologías de la seguridad de la información tienen tres objetivos fundamentales:

- **Confidencialidad o privacidad.** La confidencialidad de los datos es la protección de la información personal y sensible contra la revelación y los ataques tanto intencionados como no intencionados por parte de una posible tercera parte.
- **Autenticación.** La autenticación proporciona la seguridad de que los datos recibidos fueron en realidad enviados por quien asegura haberlo hecho. Se pueden diferenciar dos tipos de autenticación:
  - de entidad simple, ya sea del emisor o del recipiente de la información; y
  - mutua o bidireccional en la que ambos intercomunicantes se autentican uno al otro.
- **Verificabilidad.** La criptografía no sólo permite garantizar la confidencialidad y autenticidad, sino también poder corroborar incluso ante terceros que los datos recibidos fueron los originalmente emitidos y fueron emitidos por quien firmó el documento.

Además de estos objetivos básicos, y de los servicios definidos por la ISO (apartado anterior), la seguridad de un sistema debe garantizar los siguientes:

- **Disponibilidad.** Se ha de asegurar que no le sea negado a un usuario legítimo el acceso al sistema y se han de proporcionar recursos alternativos para poder utilizarlos en caso de caída de éste.
- **Idempotencia.** Cuando una operación se puede realizar un número indeterminado de veces sin que cause ningún daño al sistema, se dice que es *idempotente*.

Las tecnologías empleadas para garantizar estos servicios se pueden dividir en dos categorías:

- las relacionadas con las comunicaciones: protección de la información mientras es intercambiada; y
- las relacionadas con el sistema físico: protección de la información dentro de los ordenadores local y remoto (características de los sistemas operativos, gestión de las bases de datos, etc.).

## Métodos criptográficos

Las técnicas criptográficas, tales como el cifrado de datos o la firma digital, son empleadas en todos los sistemas que necesiten garantizar los servicios comentados en el apartado anterior. El mecanismo más básico empleado es el denominado **criptosistema** o **algoritmo criptográfico**, el cual define dos transformaciones:

- el **cifrado**: es la conversión el **texto en claro** (*plaintext*) en el **texto cifrado o criptograma** (*ciphertext*) mediante el empleo de una determinada **clave**; y
- el **descifrado**: es el proceso inverso.

La aplicación más inmediata de un algoritmo criptográfico (aunque no la única) es asegurar el servicio de confidencialidad: en lugar de transmitir el texto en claro se envía el cifrado, de forma que un atacante no podrá descifrar el contenido de la información transmitida a no ser que conozca la clave de descifrado.
La seguridad de un sistema de cifrado radica casi totalmente en la privacidad de las claves secretas. Por ello, los ataques que puede realizar un criptoanalista enemigo están orientados a descubrir dichas claves y pueden ser de varios tipos (se supone que el atacante tiene acceso al texto cifrado):

- Ataque con sólo texto cifrado (*ciphertext-only attack*).
- Ataque con texto en claro conocido (*known-plaintext attack*). El enemigo, además de poseer los criptogramas, también dispone de los textos en claro asociados.
- Ataque con texto en claro escogido (*chosen-plaintext attack*). El enemigo puede conseguir el criptograma asociado a cualquier texto en claro.
- Ataque con texto cifrado escogido (*chosen-ciphertext attack*). El enemigo puede obtener el texto en claro de cualquier criptograma.
- Ataque con texto escogido (*chosen-text attack*). Combina los dos ataques anteriormente mencionados.

## Criptografía moderna

La principal diferencia de los sistemas criptográficos modernos respecto a los clásicos está en que su seguridad no se basa en el secreto de todas las partes del procedimiento, sino en la robustez de sus operadores (algoritmos empleados) y sus protocolos (forma de usar los operadores), siendo el único secreto la clave (los operadores y protocolos son públicos).
Los algoritmos de cifrado se pueden dividir en dos categorías: simétricos o de clave privada y asimétricos o de clave pública. A continuación se explicará cada uno de ellos.

#### Métodos simétricos o de clave privada

La criptografía simétrica (e.g., DES [DES]) usa la misma clave para cifrar y para descifrar un mensaje (figura 6.1) y su seguridad se basa totalmente en el secreto de dicha clave (el algoritmo es públicamente conocido). Generalmente se utilizan dos funciones: una para realizar el cifrado y otra para el descifrado. Su principal desventaja es que hace falta que el emisor y el receptor compartan la clave.
En un buen sistema simétrico, a no ser que se conozcan todos los bits de la clave, no se podrá extraer ninguna información del texto cifrado.

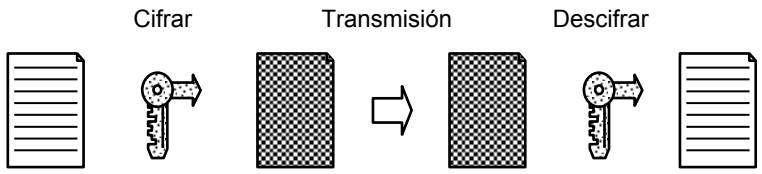



*Figura 5.1. Criptografía de clave privada.*

Dado que las claves usadas para cifrar y para descifrar son idénticas, no se pueden disociar los procesos de confidencialidad y autenticación, es decir, si se ofrece un servicio utilizando criptografía simétrica, también se está ofreciendo el otro. Entre los

algoritmos más comúnmente utilizados para ofrecer estos servicios se pueden destacar el DES (Data Encryption Standard) y el IDEA.

#### Métodos asimétricos o de clave pública

La criptografía asimétrica, introducida por W. Diffie y M. Hellman [DH76], usa dos claves, una para cifrar y otra para descifrar, relacionadas matemáticamente de tal forma que los datos cifrados por una de las dos sólo pueden ser descifrados por la otra. Cada usuario tiene dos claves, la **pública** y la **privada**, y distribuye la primera.
Este tipo de algoritmos se pueden utilizar de dos formas, dependiendo de si la clave pública se emplea como clave de cifrado o de descifrado. En el primer caso (figura 5.2), cuando un usuario, A, quiere enviar información a otro usuario, B, utiliza la clave pública de B, $Kpu_B$, para cifrar los datos. El usuario B utilizará su clave privada (que sólo él conoce), $Kpr_B$, para obtener el texto en claro a partir de la información (cifrada) recibida. Si otro usuario, C, quiere enviar información al usuario B, también empleará la clave pública $Kpu_B$. Este modo se suele emplear para proporcionar el servicio de confidencialidad, pues sólo el usuario B es capaz de descifrar los mensajes que los usuarios A y C le han enviado.

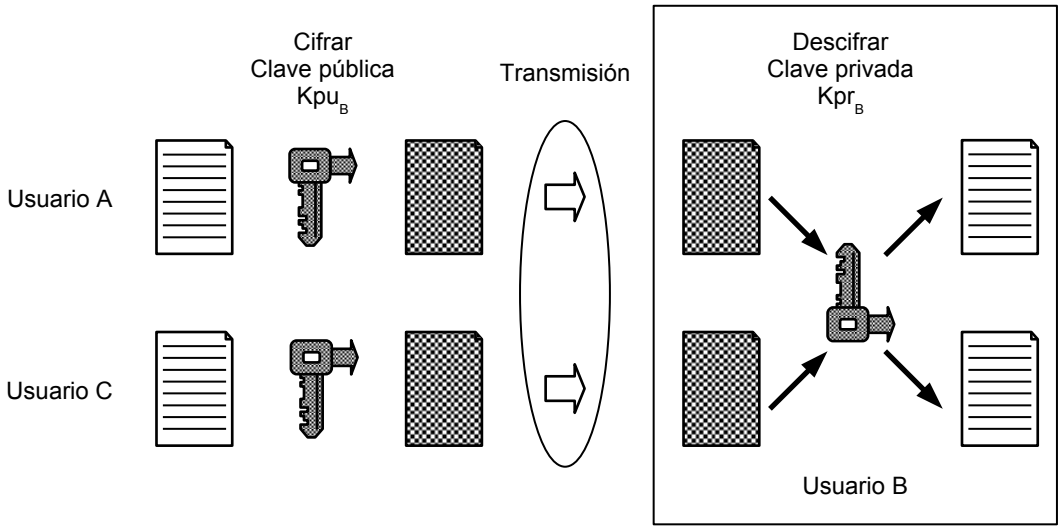



*Figura 5.2. Criptografía de clave pública: confidencialidad.*

En el otro modo de operación (figura 5.3), es el usuario B quien cifra la información utilizando su clave privada, $Kpr_B$, de forma que cualquiera que conozca $Kpu_B$ podrá descifrar la información transmitida. Este método se puede emplear para proporcionar el servicio de autenticación, ya que la obtención del texto en claro a partir del texto cifrado es una garantía de que el emisor del mensaje es el propietario de $Kpu_B$ (lógicamente, para saber que el mensaje obtenido de la descifrado del texto cifrado es el texto en claro original, éste se ha de obtener por otros medios para realizar una comparación – esto se verá más adelante). También es la base para la construcción de los mecanismos de firma digital.

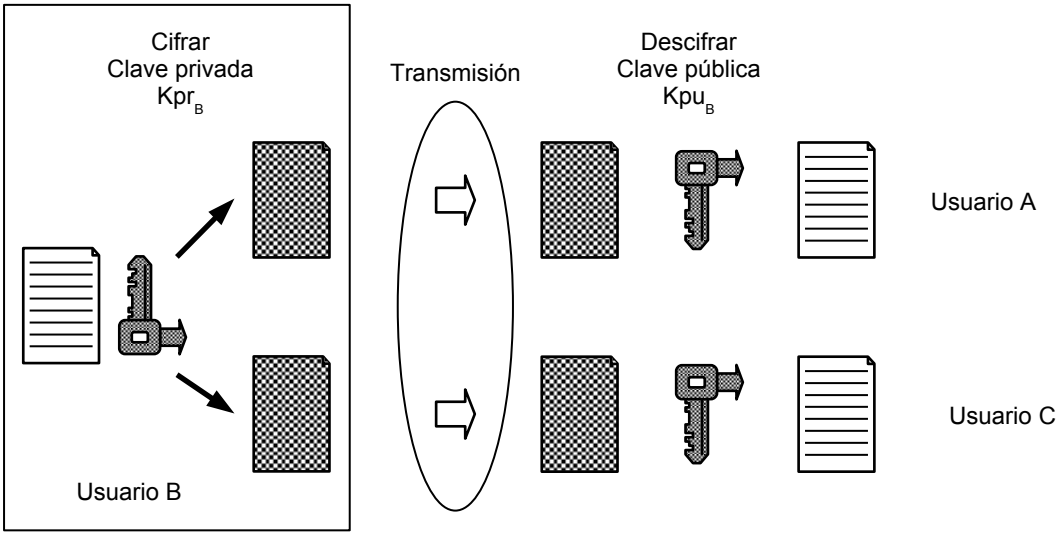



*Figura 5.3. Criptografía de clave pública: autentificación.*

Desde el punto de vista de la confidencialidad, los algoritmos de simétricos son más eficientes pues para proporcionar un mismo nivel de robustez, la carga computacional es menor que la involucrada en los cálculos asimétricos. Es por esta razón por la que en los sistemas habitualmente implementados, se emplea una combinación de algoritmos simétricos y asimétricos. Los segundos se suelen emplear para facilitar la gestión de claves simétricas utilizadas en cada sesión (session key o one-time-key) que son simétricas. Entre los algoritmos de clave pública, el más habitualmente utilizado es el RSA.

RSA

Este algoritmo fue inventado por R. Rivest, A. Shamir y L. Adleman (de sus iniciales proviene el nombre del algoritmo) [RSA78] en el Massachusetts Institute of Technology (MIT) en 1977.
RSA emplea las ventajas proporcionadas por las propiedades de los números primos cuando se aplican sobre ellos operaciones matemáticas basadas en la función módulo. Este algoritmo utiliza la función exponencial discreta para cifrar y descifrar:

$$C = M^{e} \bmod n$$
$$M = C^{d} \bmod n$$

donde $M$ es el texto en claro, $e$ es la clave pública empleada para el cifrado, $C$ es el criptograma y $d$ es la clave secreta utilizada para el descifrado.
Los cálculos matemáticos de este algoritmo emplean un número denominado el **módulo público**, $n$, que forma parte de la clave pública y que se obtiene a partir de la multiplicación de dos números primos, $p$ y $q$, diferentes y grandes (del orden de 512 bits) y que forman parte de la clave privada. La gran propiedad de RSA es que, mientras que $n$ es público, los valores de $p$ y $q$ se pueden mantener en secreto debido a la dificultad que entraña la factorización de un número grande.
La robustez del algoritmo se basa en la facilidad para encontrar dos números primos lo suficientemente grandes frente a la enorme dificultad que presenta la factorización del producto de ellos. Aunque el avance tecnológico hace que cada vez sea más rápido un posible ataque por fuerza bruta (factorizar el producto de números primos), el simple hecho de aumentar la longitud de las claves empleadas supone un incremento en la

carga computacional lo suficientemente grande para que este tipo de ataque sea inviable. Es por ello por lo que el algoritmo RSA es uno de los más empleados actualmente en las aplicaciones comerciales que necesitan realizar comunicaciones seguras. Como las funciones de cifrado y descifrado son inversas, este algoritmo, además de emplearse para cifrado, también se usa para la generación de firmas digitales, las cuales se explicarán posteriormente.

## Clasificación según el modo de operación

Los sistemas criptográficos se pueden clasificar en dos grandes grupos según el modo en el que operan: cifradores en bloque o en flujo.

## Cifradores en bloque

Un cifrador en bloque divide el mensaje, $M$, en bloques sucesivos, $M_1$, $M_2$,..., de una determinada longitud, y cifra cada uno con una clave única, $K$:

$$E_K(M) = E_K(M_1)E_K(M_2)\ldots$$

La función opera sobre un bloque de texto de tamaño fijo y genera un bloque cifrado del mismo tamaño.

### Cifrador en flujo

Un sistema de cifrado en flujo divide el mensaje, $M$, en sucesivos caracteres o bits, $m_1$, $m_2$,..., y cifra cada uno con el i-ésimo elemento de la clave, $k_i$:

$$E_K(M) = E_{k_1}(m_1)E_{k_2}(m_2)\ldots$$
$$K = k_1 k_2 \ldots$$

El algoritmo opera sobre un texto de tamaño arbitrario y genera un texto cifrado del mismo tamaño: el cifrador procesa los datos como un flujo de caracteres.
Los cifradores en bloque realizan la cifrado de la información más rápidamente que los de flujo, aunque éste no es un aspecto de demasiada importancia debido a que la velocidad de cifrado que pueden alcanzar los procesadores actuales es mucho mayor que la de los enlaces de transmisión. Además, los errores de transmisión en un criptograma no afectan al resto (a no ser que operen en modo flujo), al contrario de lo que ocurre en los de flujo, donde un error en un bit afecta a los que le siguen. Sin embargo, esto puede ser una desventaja desde el punto de vista de la seguridad. Debido a que bloques de texto en claro iguales producen criptogramas idénticos, es más fácil emplear técnicas de criptoanálisis sobre los cifradores en bloque que sobre los de flujo, en los que las repeticiones en el texto se cifran con partes diferentes de la clave. Además, la necesidad de relleno de los bloques cortos del final de los mensajes en los cifradores en bloque facilita el criptoanálisis sobre éstos.
Los sistemas de cifrado en bloque también son vulnerables a la inserción, substitución o eliminación de bloques, pues esto no afecta al resto del criptograma.
Para subsanar estos problemas de seguridad de los cifradores en bloque, se utiliza una técnica de **encadenamiento de bloques** (*block chaining*, e.g. CBC). Esto se realiza combinando el bloque $M_i$ con el anterior criptograma, $C_{i-1}$, antes del cifrado.

## Valores de control de integridad

Los criptosistemas simétricos, además de proporcionar confidencialidad, se pueden emplear también para suministrar servicios de integridad y de autenticación de origen a través de los **valores de control de integridad** (*integrity check-values*). Este mecanismo es muy similar al de CRC (*Cyclic Redundancy Check*) empleado en los

mecanismos de protección de errores, pero se utiliza una clave secreta para obtener el valor que se enviará, junto con la información a transmitir, al receptor. En recepción se comparará el valor de integridad recibido con otro obtenido de forma independiente a partir del mensaje y así poder comprobar si el texto en claro ha sido modificado durante la transmisión (figura 5.4).

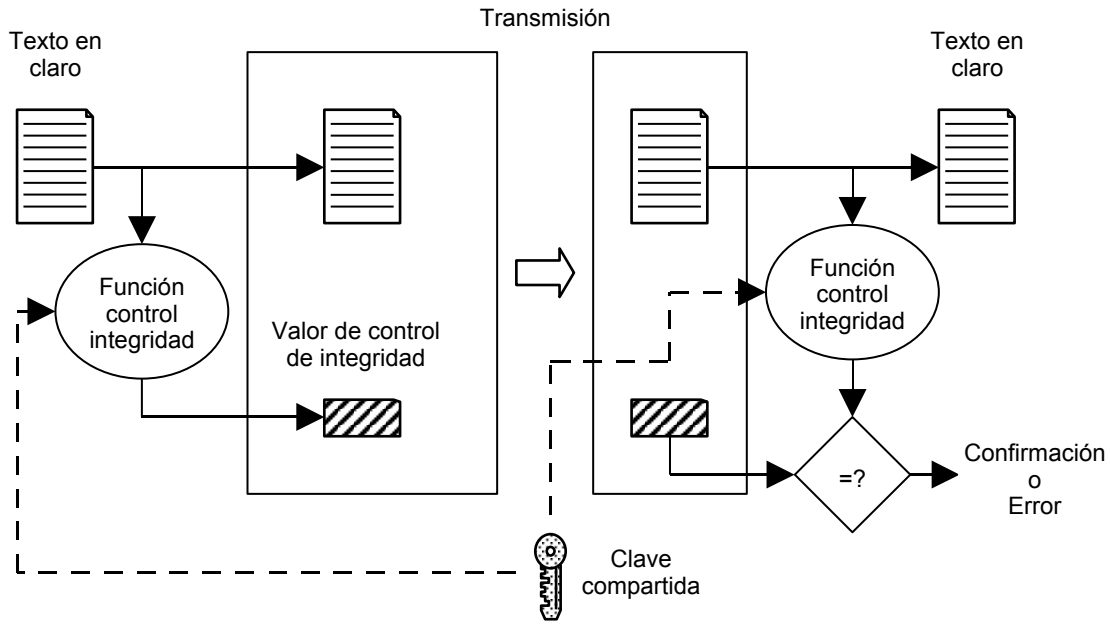



*Figura 5.4. Forma de empleo del valor de control de la integridad.*

Existe un importante mecanismo estandarizado denominado **Código de Autenticación de Mensaje** (MAC, *Message Authentication Code*) que se basa en la utilización de métodos simétricos de cifrado en bloque.
Otra forma de obtener un valor de control de integridad es aplicar una función hash a la concatenación del texto y la clave privada. A esto se lo denomina **función hash con clave** (*keyed hash function*).

## Funciones Hash

Un **valor hash** de un mensaje es un valor "único" generado a partir de él. Esto se realiza pasando el mensaje a través de una función criptográfica con las siguientes propiedades:

- Su algoritmo es conocido públicamente.
- Son de un solo sentido; o sea, a partir del valor hash no se puede obtener los datos originales.
- El valor hash es obtenido de tal forma que es muy poco probable obtener el mismo valor a partir de otros datos.

La robustez de una función hash se basa en la de las características mencionadas anteriormente. Por ejemplo, si un atacante conoce un mensaje y su valor hash y puede encontrar otros datos que produzcan el mismo valor hash, será capaz de realizar una sustitución sin que esta pueda ser detectada.
Entre los algoritmos de hash más empleados cabe destacar el SHA-1 (*Secure Hash Algorithm*) que produce una salida de 160 bits, y el MD5 de RSA Data Security Inc. que produce 128 bits de salida (considerado algo más débil que el primero).

## Firma digital

La **firma digital** es un mecanismo utilizado en los sistemas de información para asegurar la integridad del mensaje y la autenticación del emisor. Este método (figura 5.5) consiste en la obtención de un valor hash del mensaje y su posterior cifrado con la clave privada del emisor. En recepción se descifra el hash con la clave pública del emisor y se compara con otro valor hash obtenido en recepción de forma independiente a partir del mensaje recibido.

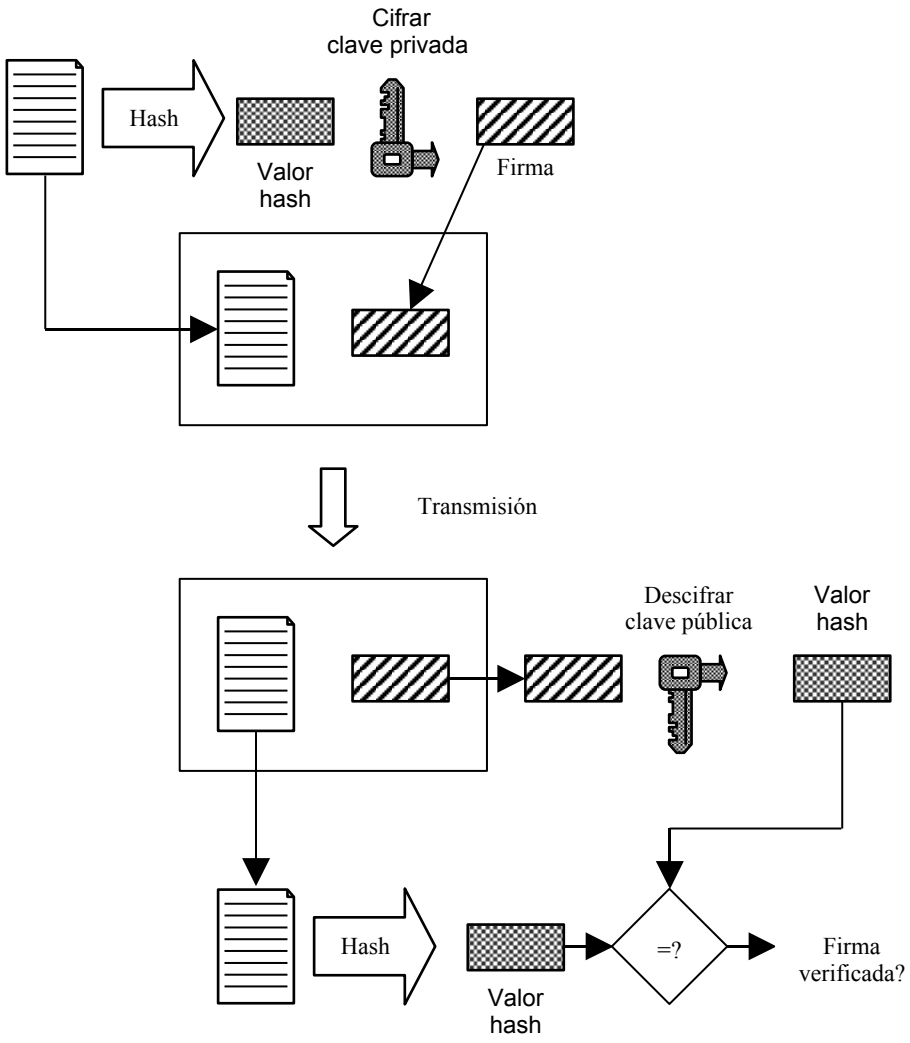



**Figura 6.5 –** Firma digital.

*Figura 5.5. Firma digital.*

La firma digital permite soportar el servicio de no repudio, ya que la verificación de la firma garantiza que ésta sólo puede haber sido generada por el poseedor de la clave privada; o sea, su usuario legítimo (a no ser que la clave privada haya sido comprometida). Esta propiedad es fundamental a la hora de realizar transacciones económicas, ya que, por ejemplo, cuando una entidad realiza un pedido de unos determinados bienes o servicios, el proveedor de éstos ha de tener un medio para poder denunciar al solicitante ante las autoridades competentes en caso de éste niegue haber realizado el mencionado pedido.

#### Digital Signature Algorithm DSA

Un algoritmo ampliamente empleado es el *Digital Signature Algorithm* (DSA) definido en el *Digital Signature Standard* (DSS), propuesto por el U.S. National Institute of Standards and Technology (NIST). Este método se basa en la función exponencial discreta en un campo de elementos finito, la cual tiene la característica de ser difícilmente reversible, ya que realizar el logaritmo discreto es una operación de una gran complejidad.
Desde el punto de vista del usuario, la creación de una firma digital se realiza de la misma manera independiente de si se emplea RSA o DSA, aunque éste último conlleva una mayor carga computacional. Sin embargo, al contrario de lo que ocurre con RSA, DSA no proporciona la capacidad para proporcionar el servicio de confidencialidad.

## Gestión de claves

Todas las técnicas criptográficas dependen en última instancia de una o varias claves, independientemente del tipo de servicio que proporcionen, por lo que la gestión de éstas es una tarea de vital importancia. Esta labor incluye básicamente:

- La generación de las claves de forma que cumplan los requisitos necesarios para su correcta utilización.
- Su distribución a todas las entidades que las puedan necesitar.
- La protección necesaria para evitar su revelación o sustitución.
- El suministro de mecanismos para informar a las entidades que las conocen en caso de que la seguridad de dichas claves haya sido comprometida.

El tipo de método empleado para llevar a cabo la gestión de las claves es diferente según el tipo de criptografía utilizada (simétrica o asimétrica).
Todas las claves tienen un tiempo determinado de vida, el **criptoperiodo**, para evitar que las técnicas de criptoanálisis tengan el suficiente tiempo e información para "romper" el algoritmo criptográfico asociado. El ciclo de vida de una clave incluye las siguientes fases:

- **Generación**. Este proceso es dependiente del algoritmo en el que se va utilizar la clave en cuestión, aunque generalmente se emplea una fuente generadora de números pseudo-aleatorios como base para la creación de la clave. El método ideal de generación sería aquél que escogiera una clave con la misma probabilidad que cualquier otra posible, ya que cualquier indicio determinista en el proceso podría facilitar el criptoanálisis.
- **Registro**. Las claves se han de enlazar a la entidad que las usará.
- **Distribución**. Aunque pueden utilizarse criptografía de clave simétrica para la distribución, habitualmente se utiliza criptografía de clave pública, utilizando el proceso que se indica en el apartado siguiente.
- **Recuperación**. Esto puede ser necesario en el caso de que una clave se pierda.
- **Reemplazo o actualización**. Esto será necesario cuando el criptoperiodo haya finalizado o en otras circunstancias especiales.
- **Revocación**. Esto se llevará a cabo cuando la seguridad de una clave haya sido comprometida.
- **Destrucción**. Esto hace referencia a borrar cualquier rastro de la clave.

## Distribución de claves simétricas mediante técnicas asimétricas

Como ya se ha explicado anteriormente, el servicio de confidencialidad se puede proporcionar utilizando tanto técnicas simétricas como asimétricas, pero estas últimas suponen una mayor carga computacional que las hace poco prácticas para la cifrado de grandes bloques de datos. Por ello, los métodos que se emplean comúnmente son una combinación de ambas categorías criptográficas.

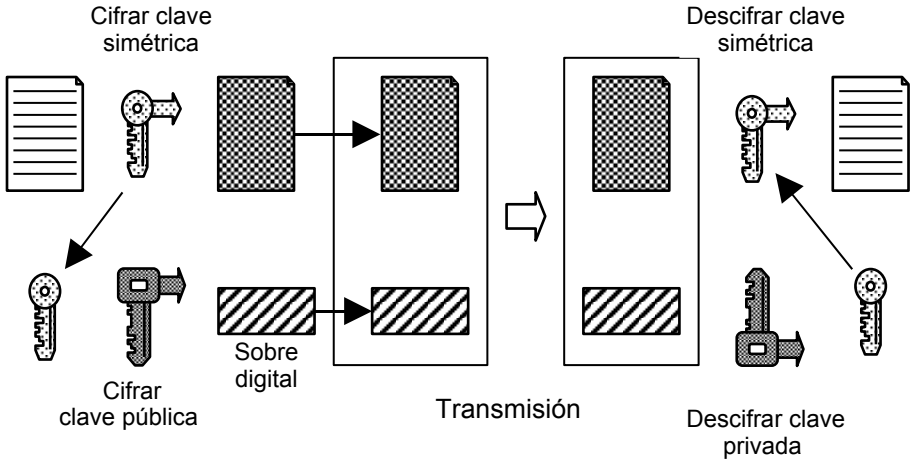



*Figura 5.6. Distribución asimétrica de claves simétricas.*

Generalmente, el proceso de distribución se realiza en los siguientes pasos (figura 5.6):

1- Se cifra el texto en claro con una clave simétrica para obtener así el texto cifrado.
2- La clave simétrica se cifra con la clave pública del recipiente. Al resultado de esta operación se lo denomina **sobre digital**.
3- Se envía el texto cifrado y la clave simétrica cifrada.
4- En recepción, se descifra la clave simétrica con la clave privada del recipiente.
5- Se descifra el texto cifrado con la clave simétrica obtenida en el paso 4, obteniendo así el texto claro original.

## Distribución de claves asimétricas

La gestión de claves asimétricas es totalmente diferente de la descrita anteriormente, en la que hace falta que cada participante almacene una clave para cada una de las otras entidades con las que mantiene comunicaciones.
En los métodos asimétricos, cada entidad sólo ha de poseer un par de claves (una privada y una pública) independientemente del número de sistemas con los que se comunique. El único requisito que se ha de cumplir es la garantía de la integridad de la clave, para así evitar que un posible atacante sustituya una clave pública y suplante a su usuario legítimo; este tipo de ataque se denomina *man-in-the-middle*. Para evitar este problema se recurre a lo que se denominan los **certificados de clave pública**, que son emitidos por unas entidades de confianza llamadas **Autoridades Certificadoras** (CAs, *Certification Authorities*) y que garantizan que una determina clave pública pertenece a su verdadero poseedor (esto será explicado en el apartado 6.4).

## Certificados

El grado de seguridad que una red telemática puede proporcionar es mayor cuando ésta se controla mediante mecanismos centralizados que cuando se hace de forma distribuida, pues una gestión global facilita la aplicación de técnicas con el objetivo de evitar ataques contra la privacidad, la integridad y la autenticación de la información. Sin embargo, aplicar un control centralizado a Internet no es viable, pues va en contra de su naturaleza. La solución actualmente empleada para securizar las comunicaciones realizadas a través de Internet se basan en métodos criptográficos asimétricos gestionados por **Terceras Partes Confiables** (TTP, *Trusted Third Parties*). Estas entidades, entre las que se encuentran las ya mencionadas CAs, permiten garantizar los servicios de confidencialidad e integridad de los datos y el no repudio de origen y destino.
Una arquitectura de gestión de certificados para Internet ha de proporcionar un conjunto de mecanismos para que la autenticación de emisores y recipientes sea simple, automática y uniforme independientemente de las políticas de certificación empleadas. La infraestructura propuesta especifica para Internet consta de una estructura jerárquica con una raíz única raíz, la IPRA (*Internet Policy Registration Authority*), que ha de definir todas las políticas globales a aplicar dentro de dicha jerarquía. La IPRA certifica a las PCA (*Policy Certification Authority*), las cuales a su vez certifican a las CAs. Las CAs son las encargadas de gestionar los certificados de los usuarios finales y deben hacer públicas sus políticas de seguridad y servicios mediante la difusión de su CPS (*Certificate Practice Statement*).
El formato más extendido para el uso de certificados es el X.509. Los campos habitualmente presentes en un certificado que sigue este formato son:

- **Versión**. Indica si la versión del certificado X.509 es la 1 (por defecto), 2 ó 3.
- **Número de serie**. El número de serie es un entero asignado por la CA emisora y que identifica unívocamente al certificado dentro del conjunto de certificados emitidos por la CA en cuestión.
- **Firma**. Identifica al algoritmo utilizado por la CA para firmar el certificado.
- **Emisor**. El nombre del emisor identifica a la entidad que ha firmado el certificado y sigue la nomenclatura de **nombres distinguibles** (DNs, *Distinguished Names*) de X.500 [X500].
- **Validez**. Indica el intervalo de tiempo en el que el certificado es válido.
- **Usuario o sujeto**. Es un nombre distinguible X.500 que identifica de forma unívoca al poseedor del certificado.
- **Información de clave pública del usuario**. Contiene la clave pública del usuario junto con el identificador del algoritmo con el que se ha de utilizar.
- **Identificadores únicos de emisor y de usuario**. Es una cadena de bits opcional que identifica al emisor o al usuario en el caso de que su DN sea reutilizado con el paso del tiempo. El IETF-PKIX WG recomienda la no reutilización de DNs y que no se empleen los identificadores únicos.
- **Campos de extensión**. Permiten la adición de nuevos campos a la estructura sin que por ello se tenga que modificar la definición ASN.1 del certificado. Cada uno de estos campos consiste en:
  - un identificador de extensión,
  - un valor para indicar si es o no crítico, y

- una codificación canónica de un valor de un tipo ASN.1 asociado con la extensión identificada.

## Generación y distribución de certificados

Las CAs tienen como misión la gestión de los denominados **certificados** (de clave pública). Un certificado está compuesto básicamente por la identidad de un usuario (subject), su clave pública, la identidad y la clave pública de la CA emisora (issuer) del certificado en cuestión, su periodo de validez y la firma digital del propio certificado. Esta firma, realizada por la CA emisora, permite que aquellas entidades que deseen realizar comunicaciones con la persona poseedora del certificado, puedan comprobar que la información que éste contiene es auténtica (suponiendo que confiemos en la CA emisora). Una vez que los certificados han sido firmados, se pueden almacenar en servidores de directorios o transmitidos por cualquier medio (seguro o no) para que estén disponibles públicamente.

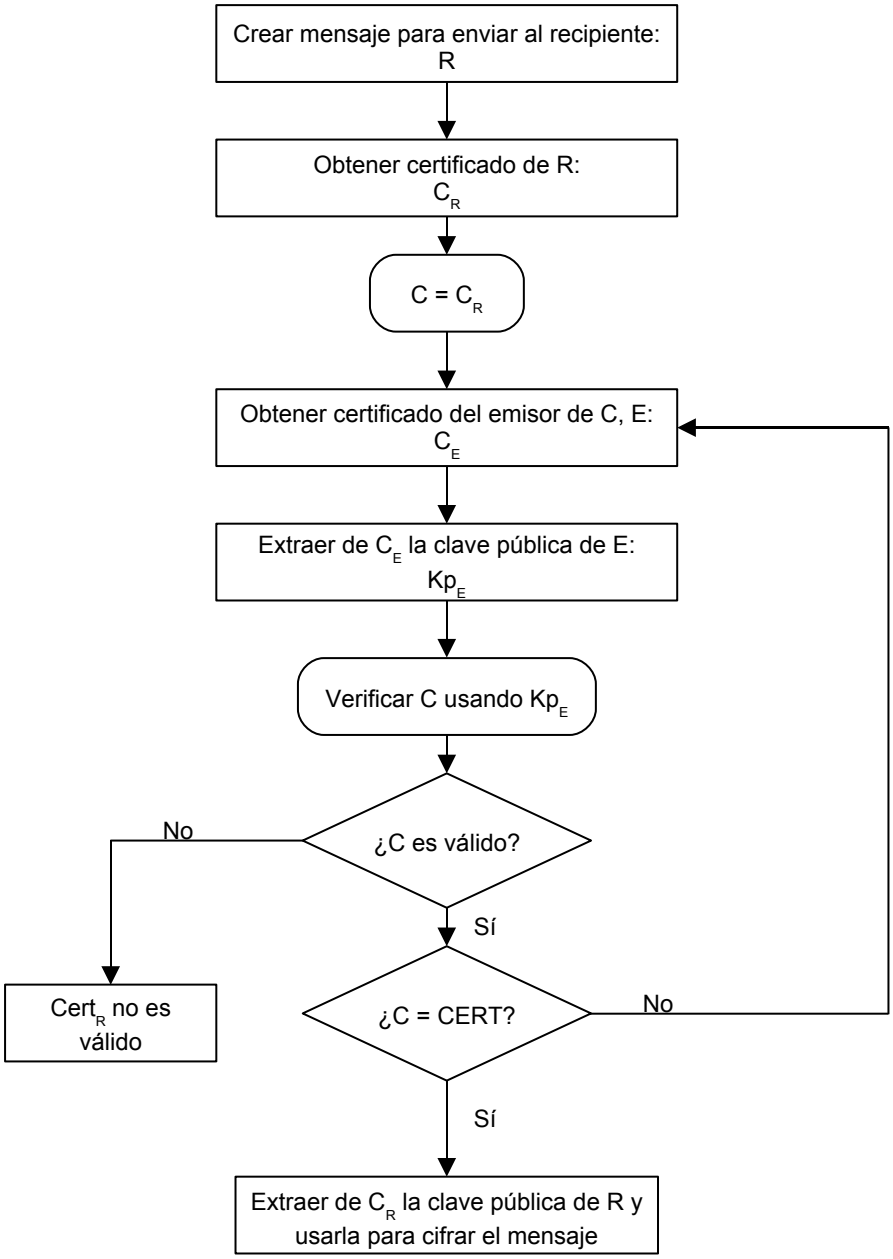



*Figura 5.7. Validación de la cadena de certificación.*

## Validación de certificados

Antes de enviar un mensaje cifrado mediante un método asimétrico, el emisor ha de obtener y verificar los certificados de los recipientes de dicho mensaje. La validación de un certificado se realiza verificando la firma digital en él incluida mediante el empleo de la clave pública de su signatario, que a su vez ha de ser validada usando el certificado correspondiente, y así sucesivamente hasta llegar a la raíz de la jerarquía de certificación. En el proceso de verificación se ha de comprobar el periodo de validez de cada certificado y que ninguno de los certificados de la cadena haya sido revocado. Esto último se realiza utilizando las CRLs (*Certificate Revocation Lists*). El esquema global de validación se muestra en la figura 5.7, donde **CERT** representa el certificado de la raíz de la jerarquía de certificación, firmado por ella misma y que se supone confiable.

Una vez validado el certificado del recipiente, se puede extraer de él la clave pública que será utilizada para realizar la cifrado. En la mayoría de los casos, esta clave se empleará para cifrar una clave de cifrado de datos (DEK, *Data Encryption Key*) que será la realmente usada para cifrar el mensaje.

## Revocación

Los certificados tienen un periodo de vida limitado, el cual está especificado en el propio certificado y que viene determinado por la política de la CA emisora. Sin embargo, en algunas ocasiones especiales la seguridad de la clave privada asociada puede haberse visto comprometida, por lo que la utilización de la correspondiente clave pública ha de ser evitada. También puede ocurrir que el propietario del certificado cambie de nombre, hecho que implica que ha de modificarse el certificado. En tales casos, la CA emisora puede **revocar** el certificado para prevenir su uso. La decisión de revocar un certificado es responsabilidad de la CA emisora, generalmente en respuesta a la petición de una entidad autorizada, como por ejemplo, el propio dueño del certificado.

## Ubicación de la Seguridad. Modelo de 4 Niveles

La Figura 6.8 muestra un par de sistemas extremos que se comunican a través de una serie de subredes. Cada una de estas subredes emplea la misma tecnología de comunicación, pudiendo ser redes de área local (LANs, *Local Area Network*) particulares, o redes de área extendida (WANs, *Wide Area Network*). Un escenario típico es un sistema extremo conectado a una LAN privada que tiene un dispositivo de interconexión (gateway) a una WAN pública. El otro sistema extremo se encuentra también en una LAN conectada también a la WAN pública.
Un sistema extremo puede soportar simultáneamente múltiples aplicaciones (correo electrónico, transferencia de ficheros, acceso a directorios,...) por uno o más usuarios simultáneos.
Los requisitos de seguridad varían con la red. Las subredes generalmente comprenden múltiples enlaces conectados a múltiples componentes de subredes y diferentes enlaces deben pasar a través de diferentes medios de seguridad. La protección de los enlaces individuales debe adecuarse a la subred.
En este escenario, distintos autores plantean un modelo de seguridad que consta de los siguientes cuatro niveles, como se muestra en la figura:

- Nivel de aplicación: Elementos de protocolos de seguridad dependientes de la aplicación.
- Nivel de sistema extremo: Elementos de protocolos de seguridad que protegen extremo a extremo.
- Nivel de subred: Elementos de protocolos de seguridad que protegen sobre una subred considerada menos fiable que el resto.
- Nivel de enlace: Elementos de protocolos de seguridad dentro de una subred concreta, que protegen un enlace que se considera menos fiable que el resto.

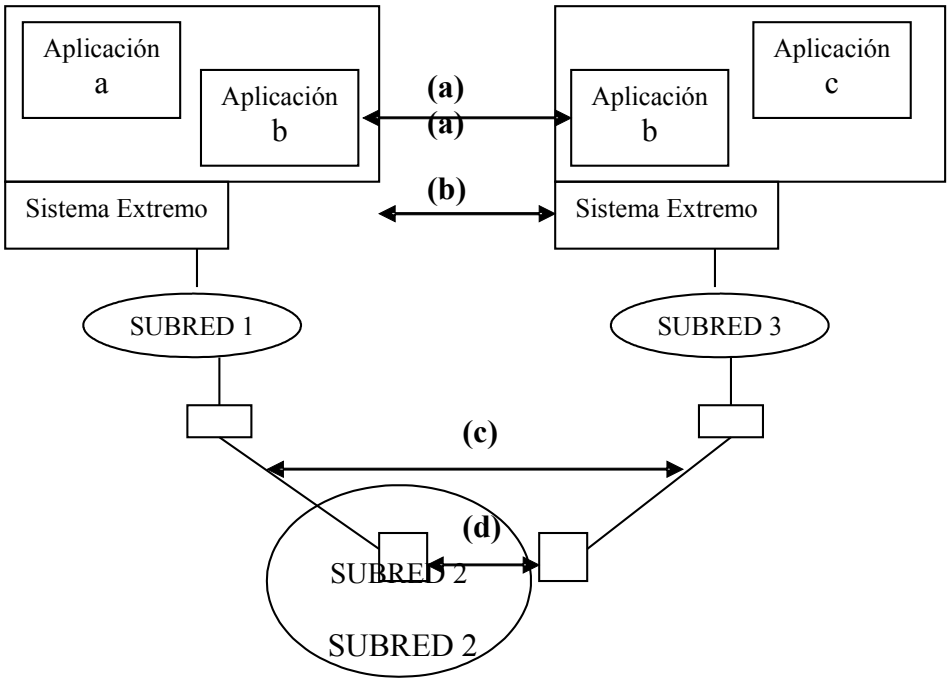



*Figura 5.8.*

La ubicación de los servicios de seguridad a niveles más altos o más bajos presenta los siguientes compromisos:

- *Mezcla de tráfico*: como resultado de la multiplexación, existe una mayor tendencia de tener datos procedentes de diferentes usuarios o aplicaciones a niveles bajos que a niveles más altos. Si se desea que usuarios y aplicaciones especifiquen individualmente la protección requerida para sus datos, entonces es preferible la ubicación de los servicios de seguridad a niveles más altos. Si se desea cierto grado de protección de toda la información independientemente de su procedencia, es preferible ubicar los servicios de seguridad a niveles más bajos.
- *Conocimiento de la ruta*: A niveles más bajos existe un mayor conocimiento de las características de seguridad de diferentes rutas y enlaces. En un medio donde tales características varían significativamente de unos enlaces a otros, la ubicación de los servicios de seguridad puede ser más efectiva y eficiente a niveles más bajos.
- *Número de puntos de protección*: La ubicación de los servicios de seguridad en el nivel más alto (nivel de aplicación), requiere una implementación de seguridad para cada aplicación en cada sistema extremo. La ubicación a niveles más bajos facilita la posibilidad de instalar los servicios de seguridad en un número menor de puntos, reduciendo el coste. La ubicación en el nivel más bajo supone la implantación de los servicios de seguridad en todos los nodos intermedios.
- *Protección de cabeceras de protocolos*: La ubicación de seguridad a niveles más altos no protege las cabeceras de los protocolos de nivel inferior. Ello permite por ejemplo ataques por análisis de tráfico. Este es uno de los motivos por el cual se suele ubicar la seguridad a niveles más bajos.
- *Asociación con origen/destino*: Algunos servicios de seguridad, como autentificación del origen de los datos o no repudio, dependen de la asociación de los datos con su origen o destino. Esta asociación se consigue más fácilmente a niveles más bajos, sobretodo el de aplicación.

Teniendo en cuenta las consideraciones anteriores, se entiende que no sea tan fácil responder a la pregunta de cual es el mejor nivel para ubicar los servicios de seguridad. Recordemos que la arquitectura de protocolos TCP/IP funciona de forma que a medida que los datos pasan de un nivel al siguiente, se va añadiendo una cabecera tal como se muestra en la Figura 5.9.

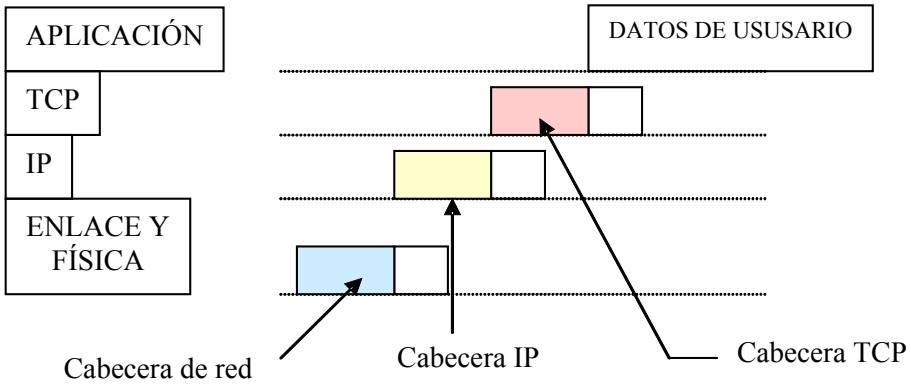



*Figura 5.9.*

Como podemos ver el número de datos a cifrar se va incrementando a medida que se baja de nivel.

## Seguridad a nivel de transporte. TLS

El protocolo TLS (*Transport Layer Security*) ha sido diseñado para proporcionar privacidad, integridad y autenticación en las comunicaciones que se realizan a través de Internet de forma que sea independiente de los protocolos de aplicación que lo utilizan, aunque típicamente se utiliza en aplicaciones Web. Las características que tienen las comunicaciones que utilizan TLS son las siguientes:

- La conexión es privada. El cifrado se utiliza después de un “handshake” inicial para definir una clave secreta. La criptografía simétrica se utiliza para el cifrado de los datos (DES, RC4, etc.)
- El uso de certificados digitales permite garantizar la identidad de sus emisores.
- La comunicación es íntegra. El transporte del mensaje incluye una comprobación de la integridad del mensaje usando una MAC cifrada. Las funciones hash seguras (SHA-1, MD5, etc.) se utilizan para cálculos MAC.

Este protocolo se basa en la especificación de SSL 3.0 [SSL] publicada por Netscape. Aunque las diferencias con éste no son grandes, sí son lo suficientemente significativas para que no pueda existir interoperabilidad entre ambos protocolos, aunque TLS incorpora un mecanismo que le permite funcionar como SSL 3.0.
TLS está compuesto por dos capas y se ubica entre el protocolo TCP y el protocolo de aplicación. Opera como una capa adicional, esto permite que el protocolo sea independiente de la aplicación, siendo posible utilizar FTP, Telnet y otras aplicaciones además de HTTP.


*Figura 5.10*

En el nivel más bajo, construido sobre un protocolo de transporte fiable (e.g., TCP) se encuentra el *TLS Record Protocol* (RP de aquí en adelante), que proporciona una conexión privada y fiable y se utiliza para encapsular otros protocolos de más alto nivel, entre los cuales se encuentra el *TLS Handshake Protocol* (HP de aquí en adelante) que permite al cliente y servidor autentificarse mutuamente y negociar un algoritmo de cifrado y claves criptográficas antes de que el protocolo de aplicación transmita o reciba el primer byte de datos.
La integridad se consigue mediante el uso de un MAC (*Message Authentication Code*) que se obtiene empleando funciones hash. El RP se puede utilizar también sin emplear MACs, pero esto sólo se suele hacer cuando otro protocolo lo está usando como un mecanismo de transporte para negociar los parámetros de seguridad.
El HP permite la autentificación del cliente y del servidor y la negociación de un algoritmo de cifrado y de las claves antes de una transacción. Proporciona una seguridad de conexión con las siguientes propiedades básicas:

- La identidad del otro participante puede autentificarse mediante el uso de criptografía asimétrica. Esto es opcional, pero generalmente hace falta la autentificación de al menos uno de ellos.
- La negociación de un secreto compartido es seguro.
- La negociación es fiable.

Los objetivos de TLS son los siguientes, en orden de preferencia:

1- **Seguridad criptográfica**: la conexión entre dos participantes ha de ser segura.
2- **Interoperabilidad**: se pueden establecer conexiones seguras entre aplicaciones diseñadas independientemente.
3- **Extensibilidad**: se pueden incorporar nuevos criptosistemas.
4- **Eficiencia**: debido a que la seguridad es el objetivo básico y a que los algoritmos criptográficos consumen muchos recursos, TLS ha incorporado un esquema de caché para reducir el número de conexiones que han de establecerse desde el principio.

## Funcionamiento y protocolos

TLS es un protocolo de capas que opera en dos fases. Funciona de la siguiente manera: TLS toma mensajes para ser transmitidos, fragmenta los datos en bloques manejables, opcionalmente comprime los datos, aplica un MAC (Message Authentication Code), cifra y transmite el resultado. Una vez los datos son recibidos, son descifrados, verificados, descomprimidos, y entonces son entregados a clientes de nivel superior.
En la primera fase, el servidor y opcionalmente el cliente, son autentificados usando certificados. Se elige un algoritmo de cifrado de entre una serie de algoritmos predefinidos, y también una clave simétrica para cada sesión de comunicación.

En la segunda fase, la principal función de TLS es cifrar y descifrar todos los datos para el protocolo de aplicación. Si es necesario, se añadirán datos de relleno para hacer que el mensaje tenga la longitud adecuada para poder aplicar el algoritmo de cifrado. Para asegurar la integridad del mensaje y su protección contra un atacante que quiera copiarlo, se pasa el mensaje junto con la clave secreta y una secuencia de números, a través de un algoritmo de hash, que produce un MAC que será enviado con el mensaje. A partir de lo explicado anteriormente, se llega a la conclusión de que los datos enviados durante la conexión podrán estar en dos formatos, Figura 5.11, dependiendo del relleno.

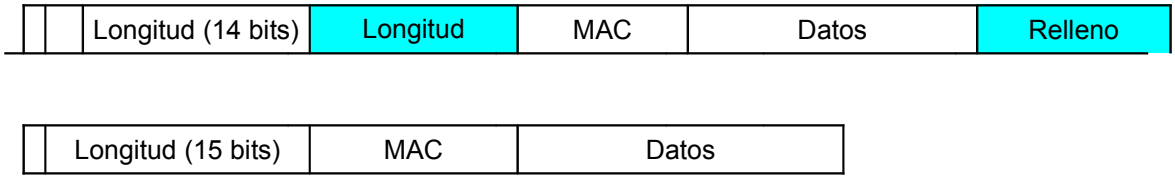



*Figura 5.11.*

## El protocolo de registro

Básicamente, el RP realiza en emisión los siguientes pasos: fragmenta el mensaje, lo comprime (opcional), le aplica un MAC, lo cifra y lo transmite. En recepción se realiza el proceso inverso: lo descifra, lo verifica, lo descomprime, los reensambla y lo transmite a la capa superior.
Se han definido cuatro protocolos clientes de esta capa: el de handshake, el de alertas, el de cambio de *cipher suite*[1] y el de datos de aplicación, pero se pueden definir otros especificando nuevos tipos de registros. Como la longitud y el tipo de registro no se protegen mediante cifrado, se ha de minimizar el valor del análisis de tráfico que de estos campos se puede obtener.
El **estado de una conexión** es el entorno en que opera el RP en cada momento e indica los algoritmos de compresión, encriptación y MAC que se están empleando.
Siempre existen cuatro estados: los de lectura y escritura activos y los de lectura y escritura pendientes. Inicialmente, los estados activos se especifican sin compresión, sin cifrado y sin MAC, y la coordinación de los estados del cliente y del servidor es responsabilidad del HP.
Los parámetros de seguridad para los estados de una conexión TLS se inicializan mediante los siguientes valores, entre otros:

- Extremo de la conexión: si la entidad opera como cliente o como servidor.
- Algoritmo de cifrado (simétrico) empleado, los parámetros que emplea.
- Algoritmo de MAC y los parámetros asociados.
- Algoritmo de compresión y parámetros asociados.
- Secreto maestro.
- Número aleatorio de cliente.
- Número aleatorio de servidor.

Una vez que se han inicializado los parámetros de seguridad se pueden activar los estados de la conexión, los cuales han de actualizarse después del procesamiento de cada registro. Cada estado de conexión incluye los siguientes elementos:

- Estado de compresión.
- Estado de cifrado.

[1] Conjunto de parámetros criptográficos.

- Secreto MAC.
- Números de secuencia de lectura y escritura.

## El protocolo de handshake

El HP permite a las entidades participantes negociar los parámetros de seguridad y activarlos, autenticarse e informar de los errores que puedan producirse. Es responsable de negociar una sesión, la cual consiste en los siguientes elementos:

- Un identificador de sesión escogido por el servidor.
- El certificado del otro participante (puede ser nulo).
- Un método de compresión.
- El cifrador a emplear, `CipherSpec`: el algoritmo simétrico de encriptación, el algoritmo MAC y cualquier parámetro relacionado.
- El secreto maestro (48 bytes).
- Un flag que indica si la sesión se puede emplear para crear nuevas conexiones.

El protocolo de handshake es uno de los clientes definidos del protocolo *TLS Record Protocol* y se utiliza para negociar los parámetros del sistema de seguridad de una sesión. Los mensajes de handshake son enviados a la capa de registro, donde son encapsulados en una o más estructuras que serán posteriormente transmitidas al otro extremo de la conexión.
La estructura `Handshake` contiene los siguientes elementos:

- El tipo del mensaje.
- La longitud en bytes del mensaje.
- El mensaje en cuestión.

### Visión global

El HP tiene lugar en los siguientes pasos:

- Intercambiar los mensajes `hello` para acordar los algoritmos, los valores aleatorios y verificar la posible reanudación de una sesión.
- Intercambiar los parámetros criptográficos necesarios para acordar un **secreto pre-maestro**.
- Intercambiar los certificados y la información criptográfica necesaria para autenticar a los participantes.
- Generar el **secreto maestro** a partir del secreto pre-maestro y los valores aleatorios.
- Suministrar los parámetros de seguridad a la capa de registro.
- Permitir al cliente y al servidor verificar que el otro participante ha calculado los mismos parámetros de seguridad y que el protocolo de handshake no ha sido atacado por un adversario.

Las capas superiores no deben confiar en que TLS negocie siempre la conexión más segura posible: un ataque de *man-in-the-middle* puede intentar que los participantes establezcan un método de comunicación menos seguro del que realmente pueden conseguir.
El handshake comienza cuando el cliente envía un mensaje `ClientHello` al cual el servidor responde con un mensaje `ServerHello` (o con uno de error, en cuyo caso la conexión no se establecerá). Estos mensajes establecen:

- La versión del protocolo.
- El identificador de sesión.
- La *cipher suite* a utilizar.
- El método de compresión.
- Además, se pueden intercambiar dos valores aleatorios: `ClientHello.random` y `ServerHello.random`, los cuales se utilizarán para las siguientes operaciones criptográficas.
- El intercambio de claves utiliza hasta cuatro mensajes: el de certificado de servidor, el de intercambio de claves de servidor, el de certificado de cliente y el de intercambio de claves de cliente. Se pueden añadir otros métodos de intercambio de claves.
- Si durante el protocolo de handshake se envía algún mensaje en un orden incorrecto, se envía un mensaje de error **fatal**.
- En la figura 5.12 se describe cómo transcurre el protocolo (* indica que el mensaje es opcional o dependiente de otros parámetros de la comunicación).

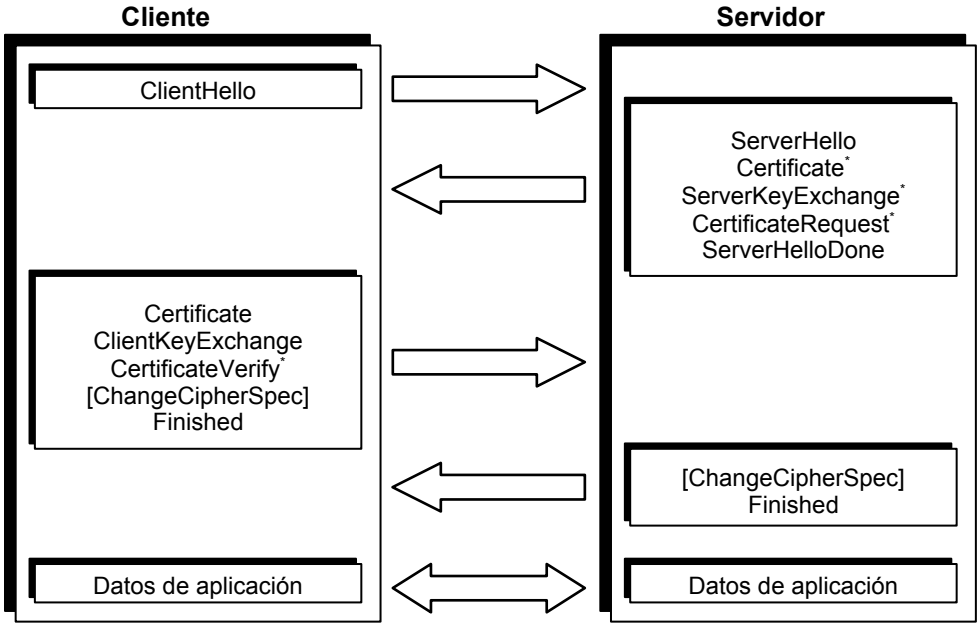



*Figura 5.12. Protocolo de handshake.*

Después del `ServerHello`, el servidor puede enviar su certificado y un mensaje de intercambio de claves. Si el servidor está autenticado, puede solicitar un certificado de cliente (si es apropiado para la *cipher suite* seleccionada). Finalmente enviará un mensaje de `ServerHelloDone`.
Si el cliente ha recibido un mensaje de solicitud de certificado, ha de enviar éste o nada en su defecto. Luego envía el mensaje de intercambio de claves. Si ha enviado un certificado con capacidad de firma, envía un mensaje de verificación de certificado firmado digitalmente. Posteriormente, manda un mensaje `ChangeCipherSpec`[2] y

[2] El mensaje `ChangeCipherSpec` no es un mensaje del protocolo de handshake.

copia el `CipherSpec` pendiente en el `CipherSpec` actual. Finalmente envía un mensaje `Finished`.
En este punto, el servidor envía su mensaje `ChangeCipherSpec`, transfiere el `CipherSpec` pendiente al actual y envía el mensaje de `Finished`.

#### Reanudar o duplicar una sesión

Cuando cliente y servidor acuerdan reanudar una sesión previa o duplicar una ya existente, el proceso se desarrolla como se describe en la figura 5.13.

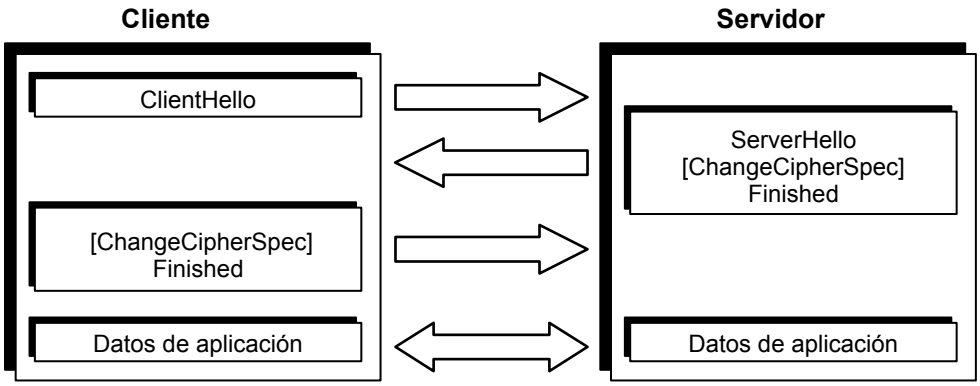



*Figura 5.13. Protocolo de handshake: reanudar o duplicar una sesión.*

El cliente envía un mensaje `ClientHello` con el identificador de sesión que quiere reanudar o duplicar. El servidor verifica dicho identificador y, si desea reanudar o duplicar la sesión, envía un `ServerHello` con el mismo ID, tras lo cual, cliente y servidor se envían mensajes `ChangeCipherSpec` y finalmente mensajes `Finished`. Si el servidor no puede verificar el identificador, envía otro nuevo y se realiza el protocolo completo.

## Seguridad a nivel de red. IPSEC (Internet Protocol Security)

En los últimos años. Internet ha experimentado un crecimiento sorprendente. Muchas organizaciones, pequeñas compañías y personas individuales, se conectan cada día. Esto ha traído dos consecuencias críticas relacionadas con la implementación del Protocolo Internet – el IPv4 – la seguridad y validez de las direcciones. Uno de los problemas de IPv4, es que utiliza direcciones de 32 bits; pero esto se solucionará con una nueva versión del protocolo que permitirá direcciones de 128 bits (IPv6). El otro problema más urgente, es la seguridad de los datos; o actualmente la falta de seguridad. IPv4 no proporciona medidas que puedan asegurar que los datos que han sido recibidos en el destino, no han sido alterados durante la transmisión; o que ellos vienen de una fuente fiable.
IPSec es un compendio de protocolos diseñados para proporcionar seguridad a las conexiones IP a través de Internet. Ha sido desarrollado por el Internet Engineering Task Force (IETF) IP Security Working Group. El objetivo del grupo de trabajo IPSec ha sido la definición de protocolos para dotar de ciertas características de seguridad de las cuales IPv4 carece. Actualmente IPSec es opcional para IPv4 y obligatorio para los desarrollos sobre IPv6 .
Podemos dividir los requisitos de seguridad en dos partes distintas:

- Autenticación & Integridad. La autenticación garantiza que los datos recibidos son los mismos que fueron enviados, y que el emisor es realmente quien dice ser. La integridad significa que podemos asegurar que los datos transmitidos han llegado a su destino sin alteraciones no detectadas. En IPSEC se consigue mediante la cabecera de autenticación (*Authentication Header – AH*)
- Confidencialidad. La confidencialidad es la propiedad de comunicarse de forma que únicamente los receptores interesados conozcan la información que ha sido enviada, y los demás individuos no puedan determinarla. IPSEC proporciona servicios de confidencialidad a través del *"Encapsulating Security Payload"* (**ESP**). ESP también puede proporcionar autenticación del origen de los datos, integridad en la conexión, y servicios anti-réplica. La confidencialidad puede ser seleccionada independientemente de los demás servicios.

Los dos mecanismos mencionados (AH y ESP) pueden usarse juntos o separados.
A continuación se proporcionan una serie de definiciones aplicables a IPSEC:

- **SPI** (Security Parameters Index): índice de parámetro de seguridad que se utiliza junto con la dirección destino (*destination address*) para identificar una asociación de seguridad (*Security Association*) en particular.
- **Security Association (SA)**: el conjunto de información sobre seguridad referido a una conexión de red dada, o grupo de conexiones.
- **Traffic Analysis**: el análisis del flujo de tráfico en la red con el propósito de deducir información que es útil para el adversario.

La cabecera de autenticación IP (AH), se ha diseñado para proporcionar a los datagramas IP integridad y autenticación, *sin confidencialidad.* La falta de confidencialidad asegura que implementaciones de la cabecera de autenticación serán ampliamente aprovechables en Internet, incluso en lugares donde la exportación, importación o uso de la criptografía para proporcionar confidencialidad está regulado, esto es debido a que los algoritmos de confidencialidad tienen problemas legales de exportación. La cabecera de autenticación soporta seguridad entre dos o más hosts implementando AH, entre dos o más gateways implementando AH, y entre un host o gateway implementando AH y una serie de hosts o gateways.
El encapsulado de seguridad de la carga útil IP (ESP), se ha diseñado para proporcionar siempre confidencialidad, y dependiendo del algoritmo y del modo, integridad y autenticación. El ESP soporta seguridad entre dos o más hosts implementando ESP, entre dos o más gateways implementando ESP, y entre un host o gateway implementando ESP y una serie de hosts o gateways.
El concepto de "***Security Association***" es fundamental tanto para la IP AH como para el IP ESP. Una asociación de seguridad está identificada unívocamente por una dirección Internet y un índice de parámetro de seguridad (SPI). La combinación de un SPI y una dirección destino únicamente identifican a una única Asociación de Seguridad. Una implementación de la AH o de ESP debe soportar este concepto de asociación de seguridad. Una asociación de seguridad normalmente incluye los parámetros que se comentan a continuación, aunque también debe incluir parámetros adicionales:

**Requeridos:**

- Algoritmo de autenticación y modo de utilización del algoritmo con la AH
- Clave(s) utilizadas con el algoritmo de autenticación que está en uso con la AH.
- Algoritmo de cifrado, modo del algoritmo y transformación que se está utilizando la IP ESP.
- Clave(s) usada con el algoritmo de cifrado que está en uso con la ESP.

- Presencia o ausencia, y tamaño de la sincronización de criptografía o inicialización del campo vector para el algoritmo de cifrado.

**Recomendados:**

- Algoritmo de autenticación y modo usado con la transformación ESP.
- Clave(s) de autenticación usada con el algoritmo de autenticación que es parte de la transformación ESP.
- Tiempo de vida de la clave o tiempo en el que se debería cambiar la clave.
- Tiempo de vida de la SA.
- Dirección(es) origen de la SA.
- Nivel de sensibilidad (seguridad) de los datos protegidos.

Una implementación AH siempre podrá usar la SPI en combinación con la dirección destino para determinar la asociación de seguridad y otros datos de seguridad relacionados, para todos los paquetes de entrada válidos.
Una asociación de seguridad es normalmente unidireccional. En una sesión de comunicación entre dos hosts, normalmente se utilizarán dos SPIs. La combinación de un SPI concreto con una dirección destino concreta, únicamente identifica a la AS.

## Modos IPSec

IPSec permite la posibilidad de creación de dos tipos de comunicación, en función de lo que queremos asegurar. Esto permitirá ocultar, o no, cierta información (direcciones IP de los nodos, protocolo utilizado, etc.) en función de nuestros intereses y también en función de los medios de que dispongamos:

**Modo Túnel:**

En este modo, los gateways proporcionan túneles para el uso de los clientes detrás de los gateways. Las máquinas de los clientes no necesitan realizar ningún proceso IPSec, todo lo que han de hacer es enrutar los paquetes hacia los gateways. En consecuencia, son los gateways de seguridad las entidades encargadas de soportar IPSec. Obviamente, es preciso realizar una traducción de direcciones en las cabeceras origen y destino del paquete original. Existe una cabecera IP externa que especifica el destino del procesamiento IPSec, además de una cabecera interna que especifica el (aparente) último destino del paquete. La cabecera del protocolo de seguridad aparece después de la cabecera IP externa y antes de la cabecera IP interna.


*Figura 5.14.*

**Modo Transporte**

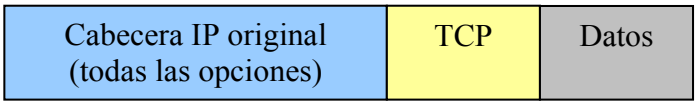
Las máquinas host, al contrario que las gateway, con implantaciones IPSec, deben también soportar el modo transporte. En este modo, el host realiza su propio proceso IPSec, y enruta los paquetes vía IPSec. En este modo, la cabecera del protocolo de seguridad aparece inmediatamente después de la cabecera IP y de cualquier opción, y antes de cualquier protocolo de nivel superior (p.e. TCP o UDP).


*Figura 5.15.*

## Cabecera de autenticación (AH: Authentication Header)

Cuando se utiliza únicamente la cabecera de autenticación no se intenta proteger el contenido, sólo asegurar su integridad. AH permite confiar en que un paquete proceda de una máquina en particular y que su contenido no ha sido alterado en el camino. Existen algunos campos de la cabecera IP que AH no puede proteger ya que éstos varían durante el camino entre el emisor y el receptor.
El servicio de autenticación puede ser proporcionado separadamente de la confidencialidad añadiendo una cabecera de autenticación AH después de la cabecera IP, pero antes de otras cabeceras en el paquete.. Los detalles se encuentran desarrollados en el RFC 2402.
Las cabeceras en un paquete están conectadas por una lista de uniones donde cada cabecera contiene un campo de "protocolo siguiente" diciendo al sistema que cabecera sigue. Las cabeceras IP tienen generalmente en este campo el valor correspondiente a TCP o UDP. Cuando se usa autenticación IPSec, la cabecera IP tiene el valor correspondiente a AH en este campo y es la cabecera de autenticación quien tiene definido en su cabecera los valores correspondientes al siguiente protocolo, ya sea TCP, UDP o IP encapsulado.
La autenticación IPSec puede ser añadida en el *modo transporte*, como una modificación del transporte IP. Esto es lo mostrado en el diagrama: La autenticación puede ser usada en *modo túnel*, encapsulando el paquete IP bajo AH y una cabecera adicional IP.

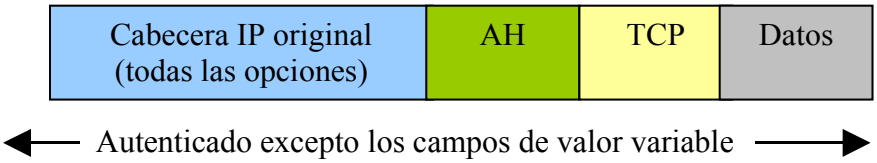
**Antes de aplicar AH**




*Ping de 1 byte (transporte)*

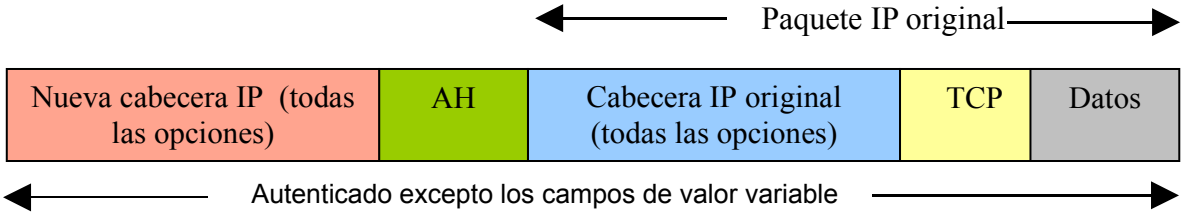
```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - -
ADDR  HEX                                                  ASCII
0000: 00 20 af 20 34 c9 00 20 af c4 35 29 08 00 45 00 | .....I...D......
0010: 00 1d 00 af 00 00 40 01 5c 30 0a 00 00 01 14 00 | ......  .*.......
0020: 00 01 08 00 ed f5 0a 0a 00 00 00 00 00 00 00 00 | .....5..........
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............
```

**Después de aplicar AH (modo transporte)**




```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - -
ADDR  HEX                                                       ASCII
0000: 00 20 af 20 34 c9 00 20 af c4 35 29 08 00 45 00 | . . 4.. ..5)..E.
0010: 00 35 00 ad 00 00 3f 33 5c e8 0a 00 00 01 14 00 | .5....?3\.......
0020: 00 01 01 04 00 00 00 00 02 00 00 00 00 02 91 21 | ..............'!
0030: a6 99 1a 1b 25 50 d5 c4 c3 17 08 00 be f6 39 09 | ....%P........9.
0040: 00 00 00
```

**Después de aplicar AH (modo túnel)**

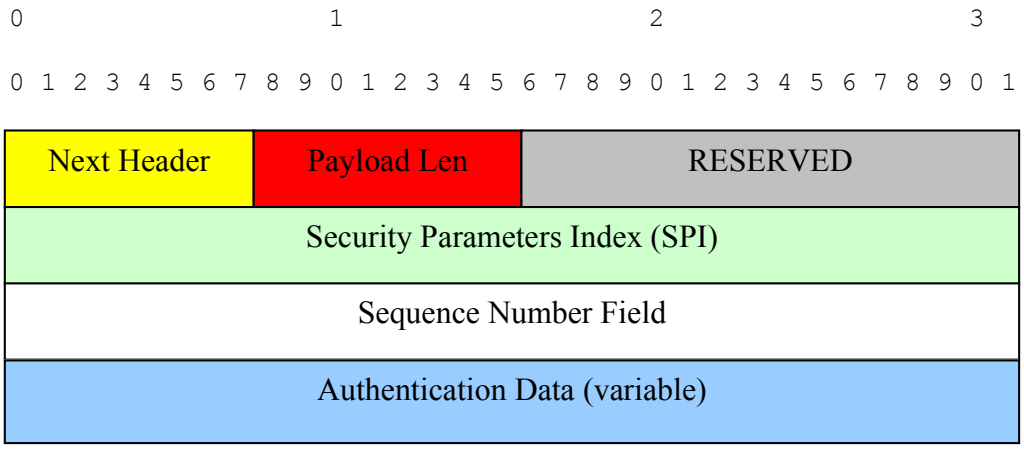



*Ping de 1 byte*

```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - -
ADDR  HEX                                                       ASCII
0000: 00 20 af 20 34 c9 00 20 af c4 35 29 08 00 45 00 | .....I...D......
0010: 00 49 00 5e 00 00 40 33 5c 23 0a 00 00 01 14 00 | ...;.. .*.......
0020: 00 01 04 04 00 00 00 00 02 00 00 00 00 02 db bb | ................
0030: 9f 73 25 1f 84 63 28 8b 24 db 45 00 00 1d 75 01 | ....d...........
0040: 00 00 1e 01 10 dd 0b 00 00 02 0c 00 00 01 08 00 | ................
0050: 90 ff 04 00 02 00 61                            | ....../
```

**Formato del Authentication Header**

La cabecera del protocolo inmediatamente antes de la cabecera AH contiene el valor **51** en caso de tratarse de IPv4.




*Ping de 1 byte*

```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - -
ADDR  HEX                                                 ASCII
0000: 00 20 af 20 34 c9 00 20 af c4 35 29 08 00 45 00 | .....I...D......
0010: 00 49 00 5e 00 00 40 33 5c 23 0a 00 00 01 14 00 | ...;.. .*.......
0020: 00 01 04 04 00 00 00 00 02 00 00 00 00 02 db bb | ................
0030: 9f 73 25 1f 84 63 28 8b 24 db 45 00 00 1d 75 01 | ....d...........
0040: 00 00 1e 01 10 dd 0b 00 00 02 0c 00 00 01 08 00 | ................
0050: 90 ff 04 00 02 00 61                            | ....../
```

**Descripción de los campos del AH**

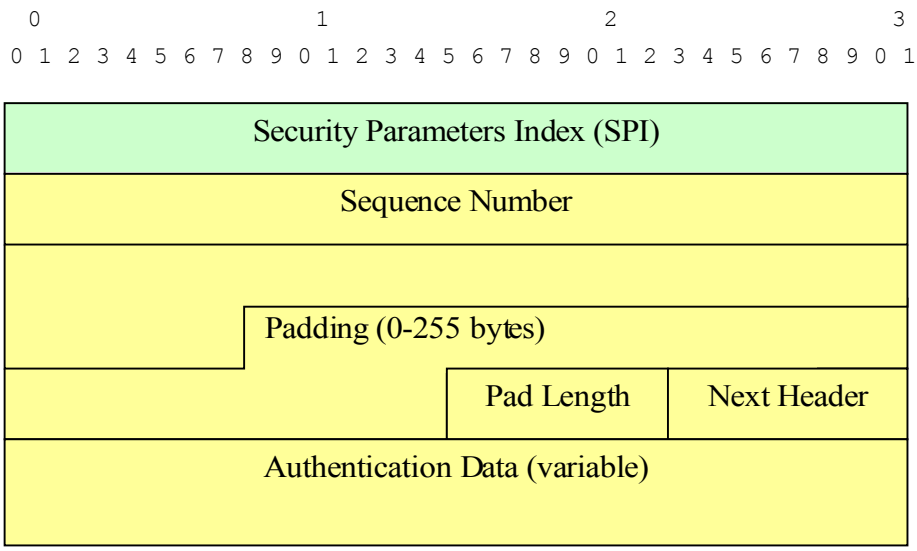
- Next Header: Es un campo de 8 bits que identifica el tipo de carga a continuación del Authentication Header.
- Payload Length: Este campo de 8 bits especifica la longitud del AH en palabras de 32 bits.
- Reserved: Este campo de 16 bits está reservado para uso futuro. El valor que debe aparecer es 0.
- Security parameters Index (SPI): El SPI es un valor arbitrario de 32 bits que, en combinación con la dirección IP destino y el protocolo de seguridad AH, identifica de forma única la Security Association para este datagrama. Los valores de SPI (de 1 a 255) han sido asignados y reservados por la IANA (Internet Assigned Numbers Authority) para uso futuro. Un valor de SPI no será asignado por la IANA hasta que éste no sea especificado en un RFC. El valor 0 de SPI se encuentra reservado para uso local y no debe ser transmitido a través de la red, su significado es el de “No Existe Asociación de Seguridad”.
- Sequence Number: Campo de 32 bits que contiene un contador incremental. El emisor debe utilizar este campo independientemente de si el receptor realiza las comprobaciones necesarias. El contador del emisor y receptor se inicializan a 0 en el momento de establecerse una SA. Una nueva SA se deberá establecer, y por tanto, resetear el contador, antes de que el contador alcance el valor $2^{32}$.
- Authentication Data: Este es un campo de longitud variable que contiene el Integrity Check Value (ICV) para este paquete. Este campo tendrá una longitud de un integral múltiplo de 32 bits.

Los datos de autenticación incluidos en la Cabecera de Autenticación IP, normalmente se calculan utilizando un algoritmo de resumen del mensaje (como MD5). Sólo aquellos algoritmos que se consideran funciones unidireccionales criptográficamente fuertes, deben ser utilizados con una cabecera de autenticación IP. Debido a que los checksums convencionales no cumplen esta condición, no deben ser utilizados con la AH.
Cuando procesamos un paquete IP de salida, para su autenticación, el primer paso consiste en que el sistema de salida localice la asociación de seguridad. La SA elegida indicará el algoritmo y su modo, la clave, y otras propiedades de seguridad que se aplicarán al paquete de salida.
Aquellos campos que cambian en el trayecto de emisor a receptor, y cuyos valores no son conocidos con certeza por el emisor, están incluidos en el cálculo de los datos de autenticación, pero son procesados de forma especial. El valor que toman estos campos es el valor cero.
El emisor debe calcular la autenticación sobre el paquete, tal como el paquete aparecerá en el receptor. El emisor coloca la salida del algoritmo resumen del mensaje calculado, en el campo de datos de autenticación con la AH.

Los campos “TIME TO LIVE” (tiempo de vida) y “HEADER CHECKSUM” (suma de comprobación) son los únicos campos de la cabecera base de IPv4 que se utilizan para el cálculo de los datos de autenticación. Para el cálculo de los datos de autenticación estos dos campos deben ser cero. Todos los demás campos de la cabecera IPv4 son procesados con su contenido actual.
La “IP Security Option” (IPSO) debe ser incluida en el cálculo de los datos de autenticación siempre que esta opción está presente en un datagrama IP. Si un sistema receptor no reconoce una opción IPv4 que está presente en el paquete, esa opción esta incluida en el cálculo de los datos de autenticación. Eso significa que cualquier paquete IPv4 que contenga una opción IPv4 que cambia durante el trayecto de una forma imprevisible por el emisor, y cuya opción IPv4 no es reconocida por el receptor, fallará la comprobación de autenticación, y consecuentemente será suspendida por el receptor.
El campo “HOP LIMIT” (limite de saltos) es el único campo de la cabecera base de IPv6 que se utiliza para el cálculo de los datos de autenticación. El valor de este campo es cero para el cálculo de los datos de autenticación. Todos los demás campos de la cabecera IPv6 deben estar incluidos en el cálculo de los datos de autenticación usando los procedimientos normales para realizar este cálculo.
Una vez recibido el paquete con la cabecera de autenticación IP, el receptor primero utiliza la dirección destino y el valor ISP para localizar la Asociación de Seguridad correcta. El receptor entonces verifica que el campo Datos de Autenticación y que el paquete de datos recibidos son consecuentes. De nuevo el campo de datos de autenticación es considerado cero con el único propósito de realizar el cálculo de autenticación. Si el procesado del algoritmo de autenticación indica que el datagrama es válido, entonces este es aceptado. En caso contrario, el receptor debe descartar el datagrama IP recibido.

## Confidencialidad (ESP: Encapsulating Security Payload)

El protocolo IPSec que proporciona confidencialidad es el ESP, Encapsulated Security Payload. El algoritmo utilizado suele ser un cifrador de bloques (habitualmente Triple DES). En las configuraciones más usuales, las claves son negociadas automáticamente, y periódicamente renegociadas, utilizando el protocolo IKE (Internet Key Exchange).
El protocolo ESP se encuentra definido en el RFC 2306. Éste proporciona servicios de confidencialidad, autenticación o ambos. Puede ser usado con o sin autenticación AH. Es importante que alguna forma de autenticación debe utilizarse cuando los datos son cifrados. Sin autenticación, la encriptación es vulnerable ante ataques activos. Por ello, ESP siempre debe incluir su propia autenticación o autenticación AH.
Formato del paquete Encapsulating Security Payload
La cabecera del Protocolo IPv4 previo a la cabecera ESP debe contener el valor 50.




```
- - - - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - - -
ADDR  HEX                                                 ASCII
0000: 00 20 af 20 34 c9 00 20 af c4 35 29 08 00 45 00 | .....I...D......
0010: 00 40 00 aa 00 00 3f 32 5c e1 0a 00 00 01 14 00 | . ......*.......
0020: 00 01 00 00 02 00 00 00 00 02 62 3e 1b 9e 86 e0 | ..............f\
0030: 20 61 79 cd ef 31 74 bf ae 06 3d 18 bd da 66 25 | ./`.............
0040: 45 7b 19 19 7e 42 e1 08 f7 bf 28 fc c2 9f       | .#..=...7...B.
```

- Security Parameters Index (SPI): El SPI es un valor arbitrario de 32 bits que, en combinación con la dirección IP de destino y el protocolo de seguridad (ESP), identifica unívocamente la Asociación de Seguridad (SA) para este datagrama. El rango de valores del SPI es de 1 a 255 y éstos se encuentran reservados por la IANA (Internet Assigned Numbers Authority) si el valor del SPI no se encuentra ya especificado en algún RFC. Normalmente el valor es determinado por el sistema destino en el momento de establecerse la SA.
- Sequence Number: Este campo de 32 bits es simplemente un contador incremental. Este campo siempre se encuentra presente aunque el receptor no utilice sus valores para el servicio anti-replay en alguna SA. Los contadores de origen y destino son puestos a cero en el momento de establecerse la SA. No se permite establecer ciclos en los valores del contador, los contadores deben ponerse a cero, estableciendo una nueva SA antes de que se llegue al paquete 232 en la transmisión.
- Payload Data: Es un campo de longitud variable que contiene los datos descritos por el campo Next Header. Si el algoritmo usado para la encriptación del Payload requiere datos de sincronización criptográfica, pe. Initialization Vector (IV), estos datos han de estar incluidos en este campo. Cualquier algoritmo de cifrado que necesite explícitamente esta sincronización por paquete, deberá incluir la longitud, cualquier estructura para estos datos y la localización de estos datos como parte de un RFC especificando como es usado este algoritmo por ESP. Si los datos de sincronización se encuentran de forma implícita, es necesario que el algoritmo para derivar los datos sea parte del RFC.
- Padding: Existen diversos factores que requieren la existencia de un campo de relleno, como el uso de algoritmo de cifrado en bloque (lo cual exige que la longitud del campo de datos a cifrar sea un múltiplo de un determinado número de bytes) y la

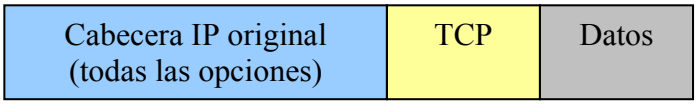
necesidad que los campos de Pad Lenght y Next Header se encuentren alineados a la derecha con una palabra de 4 bytes.

- Pad Length: Este campo indica el número de bytes de relleno utilizados en el campo anterior. El rango de valores válido es de 0 a 255, donde el valor 0 indica que no se está utilizando relleno. Este campo es obligatorio.
- Next Header: Este campo tiene una longitud de 8 bits e identifica el tipo de datos contenidos en el campo Payload Data, por ejemplo el tipo de protocolo de nivel superior incluido en el paquete. Estos valores son tomados de los IP Protocol Numbers definidos por la IANA.
- Authentication Data: Este es un campo de longitud variable que contiene el ICV (Integrity Check Value) calculado a partir del paquete ESP excepto el campo Authentication Data. La longitud de este campo viene determinada por la función de autenticación seleccionada. Este campo es opcional y por tanto sólo está incluido si el servicio de autenticación se ha seleccionado en el establecimiento de la SA. El algoritmo de autenticación seleccionado debe especificar la longitud del ICV así como las reglas de comparación y los pasos a seguir para la validación.

**Localización de la cabecera ESP**

Al igual que el Authentication Header, ESP también puede ser empleado de dos formas, modo transporte y modo túnel. En modo transporte, ESP se encuentra insertado después de la cabecera IP y antes que el protocolo de nivel superior (TCP, UDP, ICMP, etc.)

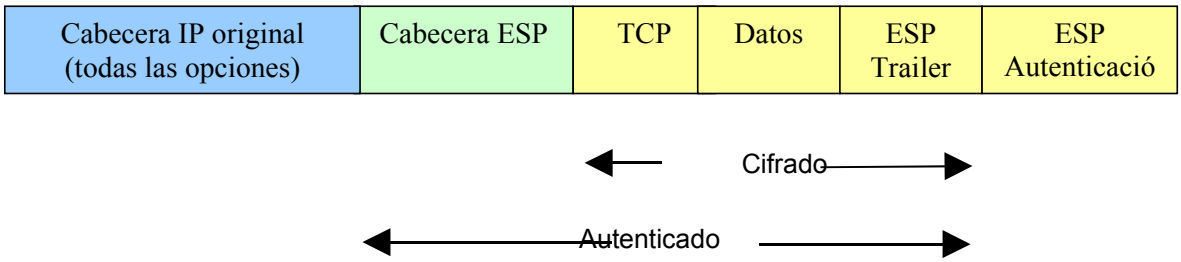
**Antes de aplicar ESP**




*Ping de 1 byte (transporte)*

```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - -
ADDR  HEX                                                  ASCII
0000: 00 20 af 20 34 c9 00 20 af c4 35 29 08 00 45 00 | .....I...D......
0010: 00 1d 00 af 00 00 40 01 5c 30 0a 00 00 01 14 00 | ...... .*.......
0020: 00 01 08 00 ed f5 0a 0a 00 00 00 00 00 00 00 00 | .....5..........
0030: 00 00 00 00 00 00 00 00 00 00 00 00             | ............
```

**Después de aplicar ESP (modo transporte)**

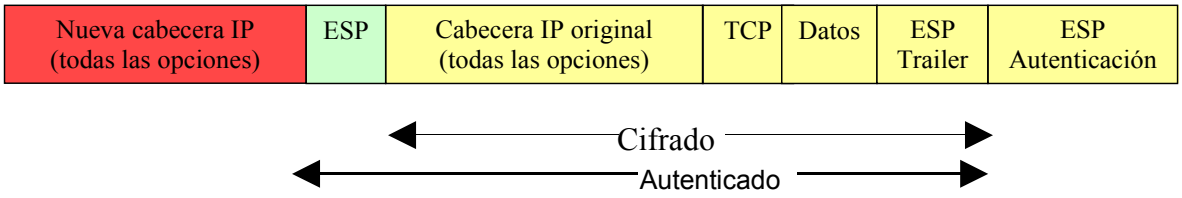


*Ping de 1 byte (transporte)*

```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - -
ADDR  HEX                                                       ASCII
0000: 00 20 af 20 34 c9 00 20 af c4 35 29 08 00 45 00 | .....I...D......
0010: 00 40 00 aa 00 00 3f 32 5c e1 0a 00 00 01 14 00 | . ......*.......
0020: 00 01 00 00 02 00 00 00 00 02 62 3e 1b 9e 86 e0 | ..............f\
0030: 20 61 79 cd ef 31 74 bf ae 06 3d 18 bd da 66 25 | ./`.............
0040: 45 7b 19 19 7e 42 e1 08 f7 bf 28 fc c2 9f       | .#..=...7...B.
```

El modo túnel puede ser empleado tanto en hosts como en gateways de seguridad. En este modo todo el paquete IP original con sus opciones, incluida la dirección de destino final, queda protegido.
La forma del paquete en modo túnel sería la siguiente:

**Después de aplicar ESP (modo túnel)**




*Ping de 1 byte (túnel)*

```
- - - - - - - - - - - - - - - - - - - - Frame 1 - - - - - - - - - - - - - - - -

ADDR  HEX                                                       ASCII
0000: 00 20 af 20 34 c9 00 20 af c4 35 29 08 00 45 00 | .....I...D......
0010: 00 50 00 a8 00 00 40 32 5b d3 0a 00 00 01 14 00 | .&.y.. .$L......
0020: 00 01 00 00 02 00 00 00 00 01 bf 73 38 35 c5 f0 | ..............E0
0030: 3f 07 ed 5a 50 b0 2a 00 3b c7 a3 63 38 e2 9b 27 | ...!&....Gt..S..
0040: e0 0c 4e 99 c7 58 fe f1 77 88 3a b5 53 26 44 2e | \.+rG..1.h......
0050: 24 c0 90 4c 2f 98 b6 d0 6e 41 90 74 a1 d4       | .{.<.q.}>...~M
```

## Autenticación más Confidencialidad

Los dos mecanismos de seguridad IP estudiados, se pueden combinar para transmitir un paquete IP que tenga autenticación y confidencialidad. Se pueden utilizar dos técnicas diferenciadas por el orden en el que se aplican los dos servicios.
La figura 5.16 muestra el caso del cifrado aplicado antes de la autenticación (modo transporte o túnel).

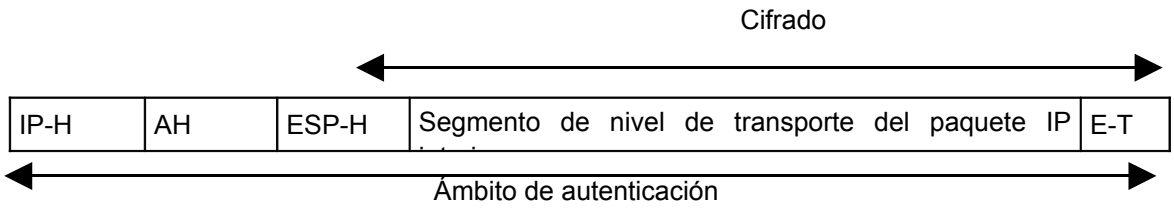



*Figura 5.16. Modelo de cifrado antes de la autenticación.*

Los campos de la trama tienen los siguientes significados.

- IP-H: Cabecera de IP más cabeceras de ampliación.
- ESP-H: Cabecera de la carga útil de seguridad de encapsulamiento.
- AH: Cabecera de autenticación.
- E-T: Campo de cierre de la carga útil de seguridad de encapsulamiento.

En este caso, el paquete IP entero transmitido se autentifica, incluyendo ambas partes, la cifrada y la no cifrada. En esta técnica el usuario primero aplica ESP a los datos que se van a proteger, después incorpora al principio la cabecera de autenticación y la(s) cabecera(s) IP en texto nativo. Existen dos subcasos:

- ESP en modo transporte: la autenticación se aplica al paquete IP entero entregado al destino, pero solamente el segmento de la capa de transporte se protege por el mecanismo de confidencialidad.
- ESP en modo túnel: la autenticación se aplica al paquete IP entero entregado a la dirección IP destino externa (por ejemplo, un cortafuegos), y la autenticación se lleva a cabo en el destino. El paquete IP interno se protege por el mecanismo de confidencialidad, para su entrega al destino IP interno.

En caso que se aplique la autenticación antes del cifrado, el resultado sería como se muestra en la figura 5.17.

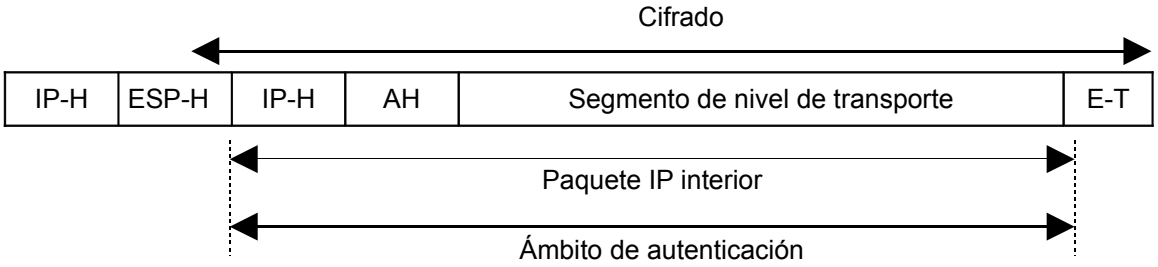



*Figura 5.17. Modelo de autenticación antes del cifrado.*

Esta técnica sólo es adecuada para ESP en modo túnel. En este caso la cabecera de autenticación se sitúa dentro del paquete IP interno. Este paquete interno es autentificado y protegido por el mecanismo de confidencialidad.
Las funciones de autenticación y cifrado se pueden aplicar en cualquier orden para ESP en modo túnel. El uso de la autenticación antes del cifrado puede ser preferible por varias razones. Primero, ya que AH se protege por ESP, es imposible que cualquiera intercepte el mensaje y altere AH sin ser detectado. Segundo, puede ser deseable almacenar la información de autenticación con el mensaje y el destino para una referencia posterior. Es más conveniente hacer esto si la información de autenticación se aplica a un mensaje no cifrado; de otra forma el mensaje tendría que ser cifrado de nuevo para verificar la información de autenticación.