UNIDAD 7 SEGURIDAD

Redes de Datos

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL SANTA FE

2025

Tabla de contenido

1. DEFINICIÓN DE SEGURIDAD DE LA INFORMACIÓN

La Seguridad de la Información, según **ISO27001** se refiere a la confidencialidad, la integridad y la disponibilidad de la información y los datos importantes para la organización. Independientemente del formato que tengan, estos pueden ser:

- Electrónicos
- En papel
- Audio y vídeo, etc.

Los gobiernos, las instituciones financieras, los hospitales y las organizaciones privadas tienen enormes cantidades de información confidencial sobre sus empleados, productos, investigación, clientes, etc. La mayor parte de esta información es reunida, tratada, almacenada y puesta a disposición de las personas que deseen revisarla.

Si se da el caso de que información confidencial de la organización, de sus clientes, de sus decisiones, de sus cuentas, caen en manos de la competencia, esta se hará pública de una forma **no autorizada** y esto puede suponer **graves consecuencias**, ya que se perderá credibilidad de los clientes, se perderán posibles negocios, se puede enfrentar a demandas e incluso puede causar la quiebra de la organización.

Es por todo esto que se convierte en una necesidad proteger la **información confidencial**, ya que es un requisito del negocio, y en muchos casos se convierte en algo ético y una obligación legal.

Para una persona normal, la **Seguridad de la Información** puede provocar un efecto muy significativo ya que puede tener diferentes consecuencias la violación de su privacidad dependiendo de la cultura de la persona.

La **Seguridad de la Información** ha crecido mucho en estos últimos tiempos. Además, ha evolucionado considerablemente y se ha convertido en una carrera acreditada mundialmente. Dentro de esta área, se ofrecen muchas especializaciones, como las siguientes:

- Planificación de la continuidad de negocio
- Ciencia forense digital
- Administración de Sistemas de Gestión de Seguridad

Realizar correctamente la **Gestión de la Seguridad de la Información** requiere establecer y mantener los programas, los controles y las políticas de seguridad que tienen la obligación de conservar la confidencialidad, la integridad y la disponibilidad de la información de la empresa.

- **Confidencialidad**: es la propiedad de prevenir que se divulgue la información a personas o sistemas no autorizados.
- **Integridad**: es la propiedad que busca proteger que se modifiquen los datos libres de forma no autorizada.
- **Disponibilidad**: es una característica, cualidad o condición de la información que se encuentra a disposición de quien tiene que acceder a esta, bien sean personas, procesos o aplicaciones.

La **Seguridad de la Información** consiste en asegurar que los recursos del Sistema de Información de una empresa se utilicen de la forma que ha sido decidido y el acceso de información se

encuentra contenida, así como controlar que la modificación solo sea posible por parte de las personas autorizadas para tal fin y por supuesto, siempre dentro de los límites de la autorización.

Los activos de información son los elementos que la **Seguridad de la Información** debe proteger. Tres son los elementos que forman los activos:

- Información: es el objeto de mayor valor para la empresa.
- Equipos: suelen ser software, hardware y la propia organización.
- Usuarios: son las personas que usan la tecnología de la organización.

Nuestro objetivo se centrará en el estudio del activo **información** circulando en la **red.**

2. SEGURIDAD EN REDES

Durante las primeras décadas de su existencia, las redes de computadoras fueron usadas principalmente por investigadores universitarios para el envío de correo electrónico, así como por empleados corporativos para compartir impresoras. En estas condiciones, la seguridad no recibió mucha atención. Pero ahora, cuando millones de ciudadanos comunes usan redes para sus transacciones bancarias, compras y declaraciones de impuestos, y que se ha encontrado una debilidad tras otra, la seguridad de las redes se ha convertido en un problema de proporciones masivas. En esta unidad analizaremos la seguridad de las redes desde varios ángulos, señalaremos muchos peligros y estudiaremos varios algoritmos y protocolos para que sean más seguras.

La seguridad es un tema amplio que cubre una multitud de pecados. En su forma más simple, se ocupa de garantizar que los curiosos no puedan leer, o peor aún, modificar en secreto mensajes dirigidos a otros destinatarios. Tiene que ver con la gente que intenta acceder a servicios remotos no autorizados. También se encarga de mecanismos para verificar que el mensaje supuestamente enviado por la autoridad fiscal que indica: “Pague el viernes o aténgase a las consecuencias” en realidad venga de ella y no de la mafia. La seguridad también se hace cargo del problema de la captura y reproducción de mensajes legítimos, y de las personas que intentan negar que enviaron ciertos mensajes.

La mayoría de los problemas de seguridad son causados intencionalmente por gente maliciosa que intenta ganar algo o hacerle daño a alguien. En la Figura 1 se muestran algunos de los tipos de transgresores más comunes. Debe quedar claro de esta lista que hacer segura una red comprende mucho más que simplemente mantenerla libre de errores de programación. Implica ser más listo que adversarios a menudo inteligentes, dedicados y a veces bien financiados.

Debe quedar claro también que las medidas para detener a los adversarios casuales tendrán poco impacto contra los adversarios serios. Los registros policiales muestran que la mayoría de los ataques no son cometidos por intrusos que interfieren una línea telefónica, sino por miembros internos resentidos. En consecuencia, los sistemas de seguridad deben diseñarse tomando en cuenta este hecho.

Los problemas de seguridad de las redes se pueden dividir en términos generales en cuatro áreas interrelacionadas: confidencialidad, autentificación, no repudio y control de integridad. La confidencialidad, también conocida como secrecía, consiste en mantener la información fuera del alcance de usuarios no autorizados. Esto es lo que normalmente viene a la mente cuando la gente piensa en la seguridad de las redes. La autentificación se encarga de determinar con quién se está hablando antes de revelar información delicada o hacer un trato de negocios. El no repudio se encarga

de las firmas: ¿cómo comprobar que su cliente en realidad hizo un pedido electrónico por 10 millones de utensilios para zurdos a 89 centavos cada uno, cuando después él afirma que el precio era de 69 centavos? O tal vez argumente que él nunca realizó ningún pedido. Por último, el control de integridad tiene que ver con la forma en que podemos estar seguros de que un mensaje recibido realmente fue el que se envió, y no algo que un adversario malicioso modificó en el camino o ideó por su propia cuenta.

Adversario	Objetivo
Estudiante	Divertirse espiando el correo electrónico de otras personas.
Cracker	Probar el sistema de seguridad de alguien; robar datos.
Rep. de ventas	Decir que representa a toda Europa y no sólo a Andorra.
Corporación	Descubrir el plan de marketing estratégico de un competidor.
Ex empleado	Vengarse por haber sido despedido.
Contador	Malversar fondos de una empresa.
Corredor de bolsa	Negar una promesa hecha a un cliente por correo electrónico.
Ladrón de identidad	Robar números de tarjetas de crédito para venderlos.
Gobierno	Conocer los secretos militares o industriales de un enemigo.
Terrorista	Robar secretos de armamento biológico.

Figura 1: Algunas personas que podrían causar problemas de seguridad y el por qué de ello.

Todos estos temas (confidencialidad, autentificación, no repudio y control de integridad) son pertinentes también en los sistemas tradicionales, pero con algunas diferencias importantes. La confidencialidad y la integridad se logran mediante el uso de correo certificado y al poner los documentos bajo llave. Ahora es más difícil robar el tren del correo de lo que era en la época de Jesse James.

Además, la gente puede por lo general distinguir entre un documento original en papel y una fotocopia, y con frecuencia esto es importante. Como prueba, haga una fotocopia de un cheque válido. Trate de cobrar el cheque original en su banco el lunes. Ahora trate de cobrar la fotocopia del cheque el martes.

Observe la diferencia de comportamiento del personal del banco. Con los cheques electrónicos, el original y la copia son idénticos. Es posible que los bancos tarden un poco en acostumbrarse a manejar esto.

La gente autentifica la identidad de otras personas por varios medios; por ejemplo, al reconocer sus caras, voces y letra. Las pruebas de firmas se manejan mediante rúbricas en papel, sellos, etc. Generalmente es posible detectar la alteración de documentos con el auxilio de expertos en escritura, papel y tinta. Ninguna de estas opciones está disponible de forma electrónica. Es obvio que se requieren otras soluciones.

Antes de adentrarnos en las soluciones, vale la pena dedicar un instante a considerar el lugar que corresponde a la seguridad de las redes en la pila de protocolos. Probablemente no haya un solo lugar. Cada capa tiene algo que contribuir. En la capa física podemos protegernos contra la intervención de las líneas de transmisión (o mejor aún, las fibras ópticas) si las encerramos en tubos sellados que contengan un gas inerte a alta presión. Cualquier intento de hacer un agujero en el tubo liberará un poco de gas, con lo cual la presión disminuirá y se disparará una alarma. Algunos sistemas militares usan esta técnica.

En la capa de enlace de datos, los paquetes de una línea punto a punto se pueden encriptar cuando se envíen desde una máquina y desencriptarse cuando lleguen a otra. Todos los detalles se pueden manejar en la capa de enlace de datos, sin necesidad de que las capas superiores se enteren de ello. Sin embargo, esta solución se viene abajo cuando los paquetes tienen que atravesar varios enrutadores, puesto que se tienen que desencriptar en cada enrutador, dentro del cual son vulnerables a posibles ataques. Además, no se contempla que algunas sesiones estén protegidas (por ejemplo, aquellas que involucren compras en línea mediante tarjeta de crédito) y otras no. Sin embargo, la encriptación de enlace (*link encryption*), como se llama a este método, se puede agregar fácilmente a cualquier red y con frecuencia es útil.

En la capa de red se pueden instalar *firewalls* para mantener los paquetes buenos e impedir que entren los paquetes malos. La seguridad de IP también funciona en esta capa. En la capa de transporte se pueden encriptar conexiones enteras de un extremo a otro; es decir, de proceso a proceso. Para lograr una máxima seguridad, se requiere que ésta sea de extremo a extremo. Por último, los asuntos como la autentificación de usuario y el no repudio sólo se pueden manejar en la capa de aplicación. Se puede ver que la seguridad no encaja por completo en ninguna capa y merece ser tratada en forma particular.

Más allá de los temas abordados en esta unidad, es necesario aclarar que numerosas fallas de seguridad en los bancos se deben a procedimientos de seguridad deficientes y a empleados incompetentes, a numerosos errores de implementación que permiten intrusiones remotas por parte de usuarios no autorizados, y a los denominados ataques de ingeniería social, en donde se engaña a los clientes para que revelen los detalles de sus cuentas. Todos estos problemas de seguridad son más frecuentes que los criminales inteligentes que intervienen líneas telefónicas y después decodifican mensajes encriptados. Si una persona pudiera entrar a cualquier sucursal de un banco con una tarjeta de débito que hubiera encontrado en la calle y solicitara un nuevo pin argumentando haber olvidado el suyo, y éste se le proporcionara en el acto (en aras de las buenas relaciones con el cliente), toda la criptografía del mundo no podría evitar el abuso. A este respecto, el libro de Ross Anderson (2008a) es una verdadera revelación, debido a que documenta cientos de ejemplos de fallas de seguridad en numerosas industrias, casi todas ellas debidas a lo que cortésmente podríamos llamar prácticas de negocios descuidadas o falta de atención a la seguridad. Sin embargo, pensamos con optimismo que a medida que el comercio electrónico se difunda, el fundamento técnico en el que se base, cuando los factores antes mencionados se cuiden bien, será la criptografía.

3. REQUISITOS DE LA SEGURIDAD

Para entender los tipos de amenazas[1] a la seguridad que existen, necesitamos partir de una definición de requisitos en seguridad. La seguridad en computadores y en redes implica cuatro requisitos:

Confidencialidad: se requiere que sólo entidades autorizadas puedan tener un acceso a la información. Este tipo de acceso incluye la impresión, la visualización y otras formas de revelado, incluyendo el simple hecho de dar a conocer la existencia de un objeto.

[1] Una amenaza es una posible causa de incidente no deseado que puede resultar en daños a un sistema u organización (ISO 27000:2009)

Integridad: se requiere que los datos sean modificados solamente por partes autorizadas. La modificación incluye la escritura, la modificación del estado, la supresión y la creación.

Disponibilidad: se requiere que los datos estén disponibles para las partes autorizadas.

Autenticación: se requiere que un computador o servicio sea capaz de verificar la identidad de un usuario.

De acuerdo con el RFC 2828 un ataque es un asalto a la seguridad del sistema derivado de una amenaza, es decir, un acto inteligente y deliberado (especialmente en el sentido de método o técnica) para eludir los servicios de seguridad y violar las políticas de seguridad de un sistema.

Una forma útil de clasificar los ataques a la seguridad (RFC 2828) es en términos de *ataques pasivos* y *ataques activos*. Un ataque pasivo intenta averiguar o hacer uso de información del sistema, pero sin afectar a los recursos de este. Un ataque activo intenta alterar los recursos del sistema o influir en su funcionamiento.

Ataques Pasivos

Los ataques pasivos consisten en escuchas o monitorizaciones de las transmisiones. La meta del oponente es la de obtener la información que está siendo transmitida, intenta conocer o hacer uso de información del sistema, pero no afecta a los recursos de este. Los ataques pasivos se dan en forma de escucha o de observación no autorizada de las transmisiones. La divulgación del contenido de un mensaje y el análisis de tráfico constituyen dos tipos de ataques pasivos.

La **divulgación del contenido de un mensaje** se entiende fácilmente. Una conversación telefónica, un mensaje de correo electrónico o un fichero transferido pueden contener información sensible o confidencial. Por ello, desearíamos impedir que un oponente averigüe el contenido de estas transmisiones.

Un segundo tipo de ataque pasivo, el **análisis de tráfico**, es más sutil. Suponga que disponemos de un medio para enmascarar el contenido de los mensajes u otro tipo de tráfico de información, de forma que, aunque los oponentes capturasen el mensaje, no podrían extraer la información del mismo. La técnica más común para enmascarar el contenido es el cifrado. Pero incluso si tenemos protección por cifrado, el oponente podría todavía observar el patrón de estos mensajes. El oponente podría determinar la localización y la identidad de los computadores que se están comunicando y observar la frecuencia y la longitud de los mensajes intercambiados. Esta información podría serle útil para averiguar la naturaleza de la comunicación que se está realizando.

Los ataques pasivos son muy difíciles de detectar, ya que no suponen la alteración de los datos. Normalmente, el tráfico de mensajes es enviado y recibido de forma aparentemente normal y ni el emisor ni el receptor son conscientes de que un tercero haya leído los mensajes u observado el patrón de tráfico. Sin embargo, es factible impedir el éxito de estos ataques, usualmente mediante cifrado. De esta manera, el énfasis en la defensa contra estos ataques se centra en la prevención en lugar de en la detección.

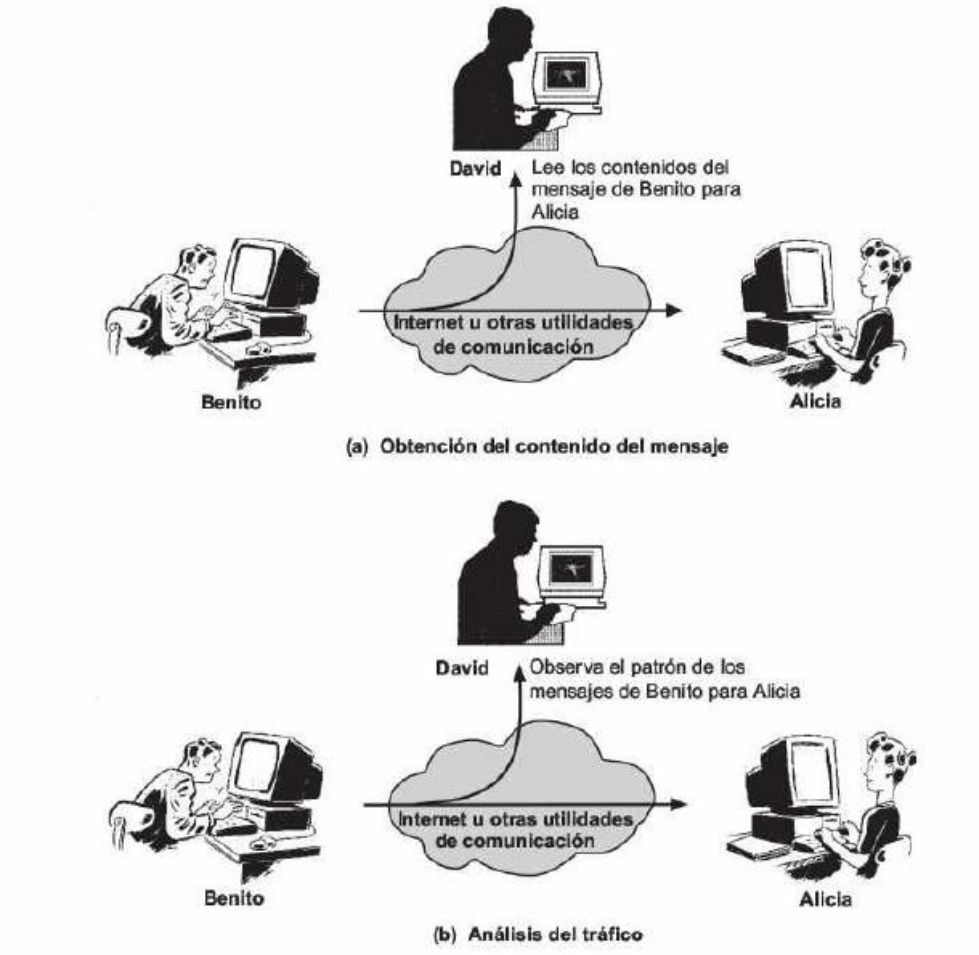

Figura 2: Ataques pasivos

Ataques Activos

Los ataques activos suponen alguna modificación del flujo de datos o la creación de flujos falsos. Siguiendo la clasificación que propone William Stallings, podemos identificar 4 categorías: suplantación de identidad, retransmisión, modificación de mensajes y denegación de servicio.

La **suplantación de identidad** tiene lugar cuando una entidad pretende ser otra entidad diferente. Un ataque de este tipo normalmente incluye una de las otras formas de ataques activos. Por ejemplo, se pueden capturar secuencias de autenticación y retransmitirlas después de que tenga lugar una secuencia válida, permitiendo así obtener privilegios correspondientes a otra.

La **retransmisión** supone la captura pasiva de unidades de datos y su retransmisión posterior para producir un efecto no autorizado.

La **modificación de mensajes** significa sencillamente que algún fragmento de un mensaje legítimo se modifica o que el mensaje se retrasa o se reordena para producir un efecto no autorizado. Por ejemplo, un mensaje con un significado «Permitir a Juan García leer el fichero confidencial de cuentas» se modifica para tener el significado «Permitir a Alfredo Castaño leer el fichero confidencial de cuentas»

La **denegación de servicio** impide o inhibe el normal uso o gestión de servicios de comunicación. Este ataque puede tener un objetivo específico. Por ejemplo, una entidad puede suprimir todos los mensajes dirigidos a un destino concreto (por ejemplo, al servicio de vigilancia de

seguridad). Otro tipo de denegación de servicio es la interrupción de un servidor o de toda una red, bien deshabilitando el servidor o sobrecargándolo con mensajes con objeto de degradar su rendimiento.

Los ataques activos presentan características opuestas a las de los ataques pasivos. Mientras que un ataque pasivo es difícil de detectar, existen medidas para impedir que tengan éxito. Por otro lado, es bastante difícil impedir ataques activos de forma ABSOLUTA. Por lo tanto, será necesario además detectarlos y recuperarse de cualquier interrupción o retardo causados por ellos. Ya que la detección tiene un efecto disuasorio, también puede contribuir a la prevención.

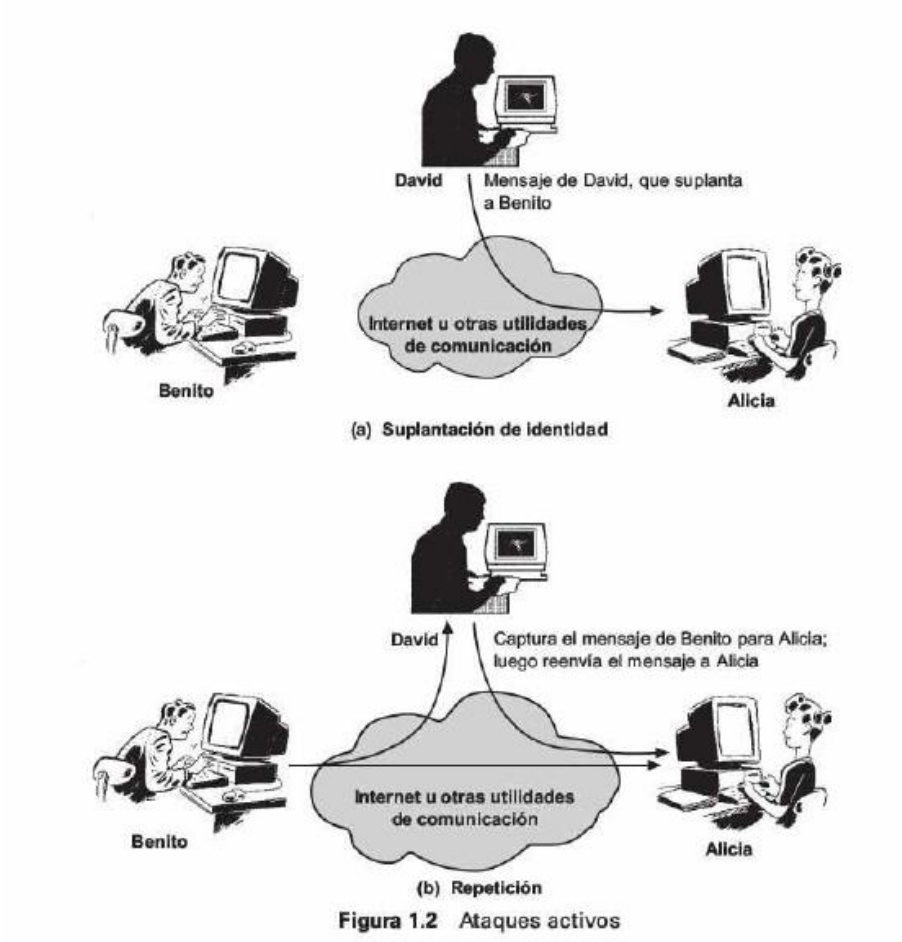

Figura 3: Ataques activos (a) Suplantación de identidad (b) Repetición

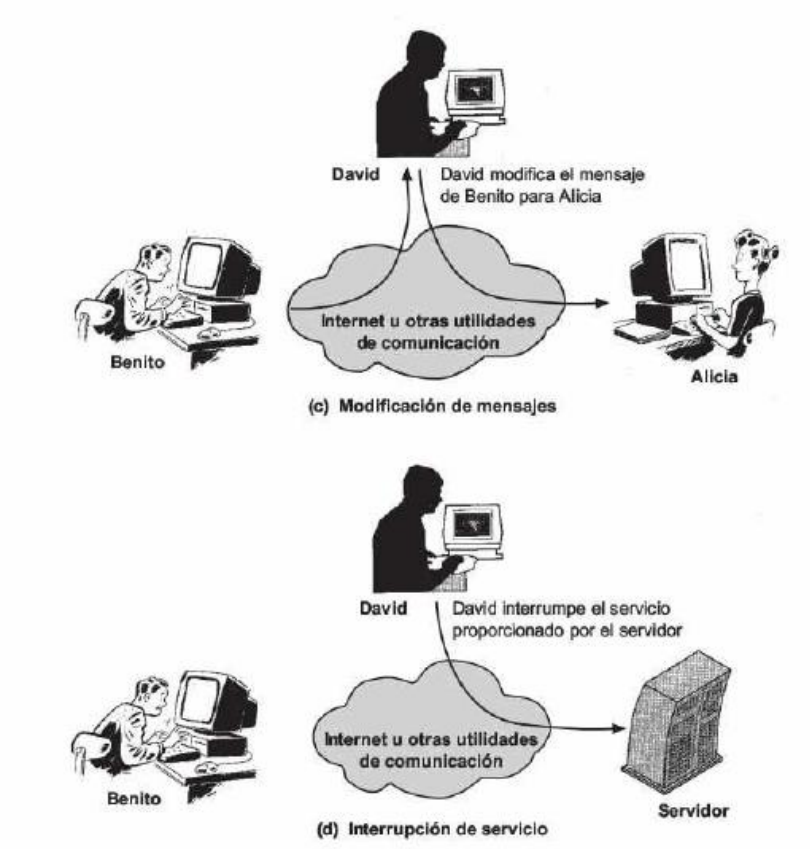

Figura 4: Más ataques activos (c) Modificación de mensajes (d) Interrupción de servicio

4. Servicios de seguridad

La recomendación X.800 define un servicio de seguridad como un servicio proporcionado por una capa de protocolo de sistemas abiertos de comunicación, que garantiza la seguridad adecuada de los sistemas o de las transferencias de datos. Quizás es más clara la definición recogida en RFC 2828: un servicio de procesamiento o de comunicación proporcionado por un sistema para dar un tipo especial de protección a los recursos del sistema; los servicios de seguridad implementan políticas de seguridad y son implementados, a su vez, por mecanismos de seguridad.

En X.800 estos servicios quedan divididos en cinco categorías, que se presentan en la tabla siguiente:

AUTENTICACIÓN	**INTEGRIDAD DE LOS DATOS**
La seguridad de que la entidad que se comunica es quien dice ser. **Autentificación de la entidad es origen/destino** Empleada conjuntamente con una conexión lógica para aportar confianza sobre la identidad de los extremos de la conexión. **Autentificación del origen de los datos** En transferencias no orientadas a la conexión garantiza que la fuente de los datos es quien dice ser.	La seguridad de que los datos recibidos son exactamente como los envió la entidad autorizada (no contienen modificación, inserción, omisión ni repetición) **Integridad de la conexión con recuperación** Proporciona integridad de los datos a todos los usuarios en una conexión y detecta cualquier modificación, inserción, omisión o repetición de cualquier dato en una secuencia completa de datos, con intento de recuperación **Integridad de la conexión sin recuperación**

CONTROL DE ACCESO La prevención del uso no autorizado de una fuente (este servicio controla quién puede tener acceso a una fuente, en qué condiciones se puede producir el acceso y qué tienen permitido los que acceden a la fuente) **CONFIDENCIALIDAD DE LOS DATOS** La protección de los datos contra la revelación no autorizada. **Confidencialidad de la conexión** La protección de los datos de todos los usuarios en una conexión **Confidencialidad no orientada a la conexión** La protección de los datos de los usuarios en un solo bloque de datos. **Confidencialidad de los campos seleccionados** La confidencialidad de campos seleccionados en los datos de usuario en una conexión o en un único bloque de datos. **Confidencialidad en el flujo de datos** La protección de la información que podría extraerse a partir de la observación del flujo del tráfico.	Igual que la anterior, pero proporciona sólo detección sin recuperación **Integridad de la conexión de campos seleccionados** Proporciona integridad en los campos seleccionados de los datos de usuario del bloque de datos transferido por una conexión y determina si los campos seleccionados han sido modificados, insertados, suprimidos o repetidos. **Integridad no orientada a la conexión** Proporciona integridad en un bloque de datos sin conexión y puede detectar la alteración de datos. Además, puede detectar una forma limitada de repetición. **Integridad no orientada a la conexión de campos seleccionados** Proporciona la integridad de los campos seleccionados en un bloque de datos sin conexión; determina si los datos seleccionados han sido modificados **NO REPUDIO** Proporciona protección contra la negación, por parte de una de las entidades implicadas en la comunicación, de haber participado en toda o en parte de la comunicación. **No repudio, origen** Prueba que el mensaje fue enviado por la parte especificada **No repudio, destino** Prueba que el mensaje fue recibido por la parte especificada

5. Criptografía

Aunque la criptografía tiene una larga historia que se remonta hasta la época de Julio Cesar, las técnicas criptográficas modernas, incluyendo muchas de las utilizadas en Internet, están basadas en los avances realizados en los últimos 30 años. El libro de Kahn, *The Codebreakers* [Kahn 1967], y el libro de Smgh, *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography* [Singh 1999], proporcionan una fascinante visión de la larga historia de la criptografía. Una exposición completa sobre critpografía requiere un libro completo [Kaufman 1995; Schneier 1995] y, por tanto, aquí solo vamos a abordar los aspectos esenciales, en particular aquellos que están en práctica en Internet.

Las técnicas criptográficas permiten a un emisor ocultar los datos de modo que los intrusos no puedan obtener ninguna información a partir de los datos interceptados. El receptor, por supuesto, deberá ser capaz de recuperar los datos originales a partir de los datos ocultados. La Figura 5 ilustra parte de la terminología más importante.

Suponga ahora que Alice quiere enviar un mensaje a Bob. El mensaje de Alice en su forma original (por ejemplo, "Bob , te quiero. Alice ") se conoce con el nombre de *texto* en claro o texto plano *(cleartext* o *plaintext)* Alice cifra su mensaje de texto en claro utilizando un algoritmo de cifrado de modo que el mensaje cifrado, que se conoce con el nombre de texto cifrado *(ciphertext),* es ininteligible para cualquier intruso. Es interesante observar que, en muchos sistemas criptográficos modernos, incluyendo los utilizados en Internet, la propia técnica de cifrado es *conocida*, en el sentido de que es pública, está estandarizada y está disponible para todo el mundo (por ejemplo, [RFC 1321; RFC 2437; RFC 2420; NIST 2001]), incluso para los potenciales intrusos). Evidentemente, si todo el mundo conoce el método utilizado para codificar los datos, entonces deberá existir algún tipo de información secreta que impida a un intruso descifrar los datos transmitidos: aquí es donde entran en acción las claves.

En la Figura 5 Alice proporciona una clave, K_A, una cadena de números o caracteres como entrada para el algoritmo de cifrado. El algoritmo de cifrado toma la clave y el mensaje de texto en claro, *m,* como entrada y genera el texto cifrado como salida. La notación $K_A(m)$ hace referencia al formato en texto cifrado (cifrado utilizando la clave K_A) correspondiente al mensaje de texto en claro, *m.* El algoritmo de cifrado real con el que se vaya a utilizar la clave K_A resultará evidente dentro del contexto. De forma similar, Bob proporcionará una clave, K_B, al algoritmo de descifrado, que toma el texto cifrado y la clave de Bob como entrada y genera como salida el texto en claro original. En otras palabras, si Bob recibe un mensaje cifrado $K_A(m)$, lo descifra realizando el cálculo $K_B(K_A(m)) = m$. En los sistemas de clave simétrica, las claves de Alice y de Bob son idénticas y deben mantenerse en secreto. En los sistemas de clave pública, se emplea una pareja de claves. Una de las claves es conocida tanto por Bob como por Alice (de hecho, es conocida por todo el mundo). La otra clave solo es conocida por Bob o por Alice, pero no por ambos. En las siguientes subsecciones vamos a estudiar los sistemas de clave simétrica y de clave pública más detalladamente.

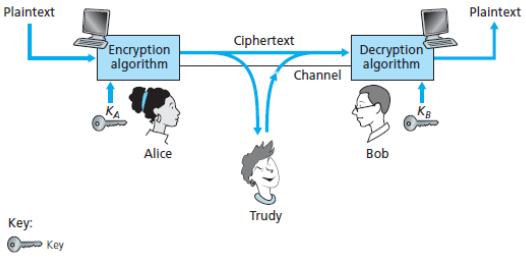

Figura 5: Componentes criptográficos

6. Cifrado simétrico

El cifrado simétrico, también denominado cifrado convencional o de clave única, era el único tipo de cifrado en uso antes de la introducción del cifrado de clave pública a finales de la década de los setenta. Innumerables individuos y grupos, desde Julio César, pasando por la fuerza alemana Uboat, hasta los actuales usuarios diplomáticos, militares y comerciales, han empleado el cifrado simétrico para la comunicación secreta. De los dos tipos de cifrado, es todavía el más utilizado.

Un esquema de cifrado simétrico tiene cinco ingredientes:

Texto nativo (*plaintext*): es el mensaje original o datos que se proporcionan como entrada del algoritmo.

Algoritmo de cifrado: el algoritmo de cifrado lleva a cabo varias sustituciones y transformaciones sobre el texto nativo.

Clave secreta: la clave secreta es también una entrada del algoritmo de cifrado. Las sustituciones y transformaciones concretas realizadas por el algoritmo dependen de la clave.

Texto cifrado (*ciphertext*): es el mensaje alterado que se produce como salida. Depende del texto nativo y de la clave secreta. Para un mensaje dado, dos claves diferentes producen dos textos cifrados diferentes.

Algoritmo de descifrado: es esencialmente el algoritmo de cifrado ejecutado a la inversa. Toma como entradas el texto cifrado y la clave secreta y produce como salida el texto nativo original.

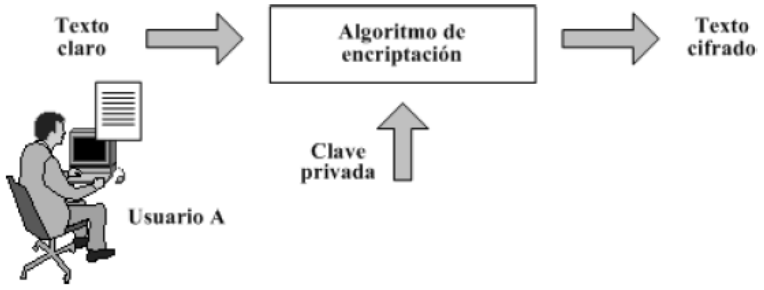

Figura 6: Encriptación mediante un algoritmo simétrico

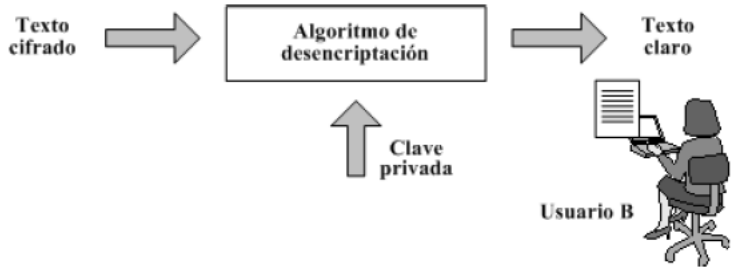

Figura 7: Desencriptación mediante algoritmo simétrico (la clave privada es la misma que se ha utilizado para cifrar el mensaje)

Existen dos requisitos para la utilización segura del cifrado simétrico:

1. Se necesita un algoritmo de cifrado robusto. Como mínimo, es de desear que el algoritmo cumpla que, aunque un oponente conozca el algoritmo y tenga acceso a uno o más textos cifrados, sea incapaz de descifrar el texto o averiguar la clave. Este requisito se suele enunciar de una forma más estricta: el oponente debería ser incapaz de descifrar el texto o descubrir la clave incluso si él o ella poseyera varios textos cifrados junto a sus correspondientes textos nativos.

2. El emisor y el receptor tienen que haber obtenido las copias de la clave secreta de una forma segura y deben mantenerla en secreto. Si alguien puede descubrir la clave y conoce el algoritmo, toda comunicación que utilice esta clave puede ser leída.

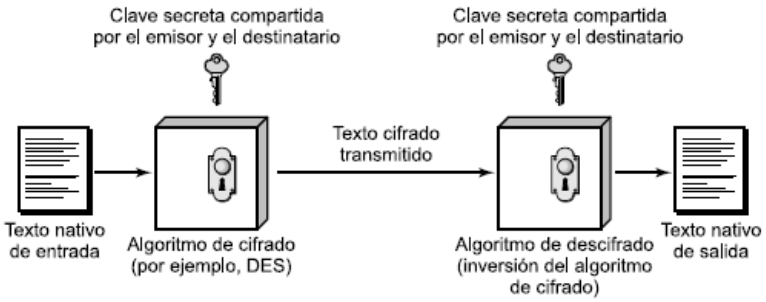

Figura 8: Modelo simplificado de cifrado simétrico

Los sistemas criptográficos simétricos pueden tener dos formas de funcionamiento:

• **Cifrado en bloque o poligráfico (en idioma inglés, block cipher):** el mismo algoritmo de encriptación se aplica a un bloque de información (grupo de caracteres o número de bytes) repetidas veces, usando la misma clave. De este modo, es posible combinar varias sustituciones y transposiciones. Cuando realiza cifrado, una unidad de cifrado por bloques toma un bloque de *texto plano* o claro como entrada y produce un bloque de igual tamaño de texto cifrado. La transformación exacta es controlada utilizando una segunda entrada: la clave secreta. El descifrado es similar: se ingresan bloques de texto cifrado y se producen bloques de texto plano.

El más sencillo es el modo *electronic codebook* (ECB), en el cual los mensajes se dividen en bloques y cada uno de ellos es cifrado por separado utilizando la misma clave K. En las aplicaciones de redes de computadoras, normalmente es necesario cifrar mensajes de gran tamaño (o largos flujos de datos). Si aplicamos un cifrado de bloques como el descrito, descomponiendo simplemente el mensaje en bloques de k bits y cifrando independientemente cada bloque, aparece un problema bastante sutil pero de gran importancia. Para ver en qué consiste, observe que dos o más de los bloques del texto en claro podrían ser idénticos. Por ejemplo, el texto en claro en dos o más bloques podría ser “HTTP/1.1”. Para estos bloques idénticos, el cifrado de bloque produciría, por supuesto, el mismo texto cifrado. De ese modo, un atacante podría posiblemente adivinar el texto en claro cuando viera bloques de texto cifrado idénticos y podría incluso ser capaz de descifrar el mensaje completo identificando bloques de texto cifrado idénticos y utilizando el conocimiento acerca de la estructura de protocolos subyacente [Kaufman 1995]. Por ello los sistemas de cifrado de bloque suelen utilizar una técnica denominada Encadenamiento de bloques cifrados (CBC, Cipher Block Chaining). Específicamente, CBC opera como sigue:

1. Antes de cifrar el mensaje (o el flujo de datos), el emisor genera una cadena aleatoria de bits, denominada Vector de inicializacion (IV, Initialization Vector). Denotaremos a este vector de inicializacion mediante c(0). El emisor envía el vector IV al receptor sin cifrar.
2. Para el primer bloque, el emisor calcula m(1) XOR c(0), es decir, calcula la operación OR exclusiva del primer bloque de texto en claro con IV. A continuación, introduce el resultado en el algoritmo de cifrado de bloque, para obtener el correspondiente bloque de texto cifrado; es decir, c(1) = Ks(m(1) XOR c(0)). El emisor envía después el bloque cifrado c(1) al receptor.
3. Para el i-ésimo bloque, el emisor genera el i-ésimo bloque de texto cifrado utilizando la formula c(i) =| Ks(m(i) XOR c(i — 1)).

Examinemos ahora algunas de las consecuencias de este método. En primer lugar, el receptor continuará pudiendo recuperar el mensaje original. De hecho, cuando el receptor reciba c(i), descifrará el mensaje con Ks para obtener s(i) = m(i) XOR c(i - 1); puesto que el receptor también conoce c(i - 1), puede entonces obtener el bloque de texto en claro a partir de la formula m(i) = s(i) xor c(i - 1). En segundo lugar, incluso si dos bloques de texto en claro son idénticos, los textos cifrados correspondientes serán (casi siempre) diferentes. En tercer lugar, aunque el emisor envíe el vector IV sin cifrar, ningún intruso podrá descifrar los bloques de texto cifrado dado que no conocen la clave secreta, S. Por último, el emisor solo envía un bloque de sobrecarga (el vector IV), con lo que el uso de ancho de banda sólo se incrementa de una forma prácticamente despreciable, asumiendo que estemos utilizando mensajes de gran longitud (compuestos de centenares de bloques).

En la actualidad suele trabajarse con bloques de bits, ya que los mensajes a encriptar se codifican previamente mediante bits (utilizando, por ejemplo, ASCII). Puede resultar necesario aplicar el mecanismo conocido como padding para completar algunos bloques de un determinado mensaje con bits adicionales hasta alcanzar el tamaño del bloque con el que trabaja el algoritmo. Entre los algoritmos de cifrado en bloque, los más conocidos son DES, IDEA, AES).

La técnica CBC tiene una importante consecuencia a la hora de diseñar protocolos de red seguros: necesitaremos proporcionar un mecanismo dentro del protocolo para transferir el vector IV desde el emisor al receptor. Posteriormente en el capítulo veremos cómo se hace esto en diversos prtocolos.

- **Cifrado en flujo, bit a bit, o byte a byte (en idioma inglés, stream cipher):** el algoritmo de encriptación se aplica a un elemento de información (carácter, bit) mediante un flujo que constituye la clave y, en teoría, es aleatorio y de un tamaño mayor que el mensaje. Para generar el flujo que constituye la clave, se emplea un generador de secuencias pseudoaleatorias y un circuito electrónico conocido como Registro de Desplazamiento Lineal; por este motivo, estos algoritmos resultan muy eficientes si se implementan mediante hardware especializado. En este tipo de algoritmos sólo se realizan sustituciones, mediante una operación XOR entre cada bit de información y cada bit de la secuencia que forma la clave. Se emplean en situaciones donde son altamente probables los errores de transmisión, ya que de este modo no se propagan los errores; además, presentan la ventaja, frente a los sistemas de cifrado en bloque, de que la información puede encriptarse o desencriptarse sin tener que esperar a que se complete un bloque de un determinado -tamaño de bits, por lo que son especialmente apropiados para los sistemas de comunicaciones en tiempo real (tal como telefonía móvil digital). Entre los algoritmos más conocidos, pueden mencionarse RC4 o A5, éste último empleado en la telefonía digital GSM.

Ataques al cifrado simétrico

Existen dos enfoques generales para atacar el esquema de cifrado simétrico. El primer ataque se conoce como **criptoanálisis**. Los ataques de criptoanálisis se basan en la naturaleza del algoritmo junto a algún posible conocimiento de las características generales del texto nativo o incluso de algunas partes de texto nativo y cifrado. Este tipo de ataque explota las características del algoritmo para intentar deducir un texto nativo concreto o deducir la clave que se esté utilizando. Si el ataque tiene éxito en la deducción de la clave, el efecto es catastrófico: todos los mensajes cifrados con esa clave, pasados y futuros están comprometidos.

El segundo método, conocido como ataque por **fuerza bruta**, consiste en probar cada posible clave sobre un fragmento de texto cifrado hasta que se obtenga una traducción inteligible de texto nativo. El rendimiento de este ataque es inversamente proporcional al tamaño de las claves y directamente proporcional a la capacidad de procesamiento. Es por ello por lo que, en la medida que se mejoran las velocidades de procesamiento se deben aumentar los tamaños de clave.

Distribución de claves simétricas

Para que el cifrado simétrico funcione, las dos partes deben tener la misma clave para realizar un intercambio seguro, y esa clave debe protegerse del acceso de otros. Más aún, es deseable cambiar frecuentemente la clave para limitar la cantidad de datos comprometidos si un atacante la descubre. Por lo tanto, la robustez de un sistema criptográfico depende de la técnica de distribución de claves, término que se refiere a la manera de entregar una clave a dos partes que desean intercambiar datos, sin permitir que otros vean dicha clave.

La distribución de claves se puede conseguir de diferentes maneras. Para dos partes *A y B:*

1. Una clave podría ser elegida por *A y* entregada físicamente a *B.*
2. Una tercera parte podría elegir la clave y entregarla físicamente a *A y* a *B.*
3. Si con anterioridad *A y B* han estado usando una clave, una parte podría transmitir la nueva clave a la otra cifrada usando la antigua.
4. Si *A y B* disponen de una conexión cifrada a una tercera parte *C, C* podría distribuir mediante los enlaces cifrados una clave a *A y* a *B.*

Las opciones 1 y 2 implican la entrega manual de una clave. Para el cifrado de enlace es un requisito razonable, porque cada dispositivo de cifrado de enlace solamente intercambia datos con su interlocutor en el otro lado del enlace. Sin embargo, para cifrado extremo a extremo la entrega manual es difícil. En un sistema distribuido, cualquier *host* o terminal podría necesitar demasiado tiempo en ocuparse de los intercambios de clave con muchos otros *hosts* o terminales. Por eso, cada dispositivo necesita un número de claves, suministradas de forma dinámica. El problema es especialmente difícil en un sistema distribuido de área ancha.

La opción 3 es una posibilidad tanto para cifrado de enlace como para cifrado extremo a extremo, pero si un atacante consiguiera acceso a una clave, se revelarían todas las subsiguientes. Aunque se realizaran cambios frecuentes de las claves de cifrado de enlace, deberían hacerse manualmente. Para suministrar claves para el cifrado extremo a extremo, es preferible la opción 4.

La Figura 9 ilustra una implementación que satisface la opción 4 para cifrado extremo a extremo. En la figura se ignora el cifrado de enlace. Éste se puede añadir, o no, según se requiera. Para este esquema se identifican dos tipos de claves:

Clave de sesión: cuando dos sistemas finales *(hosts,* terminales, etc.) desean comunicarse, establecen una conexión lógica (por ejemplo, circuito virtual). Durante el tiempo que está activa esa conexión lógica, todos los datos de usuario se cifran con una clave de sesión, válida sólo para esa conexión. Al finalizar la conexión o sesión, la clave es destruida.

Clave permanente: una clave permanente es una clave usada entre entidades con el propósito de distribuir claves de sesión.

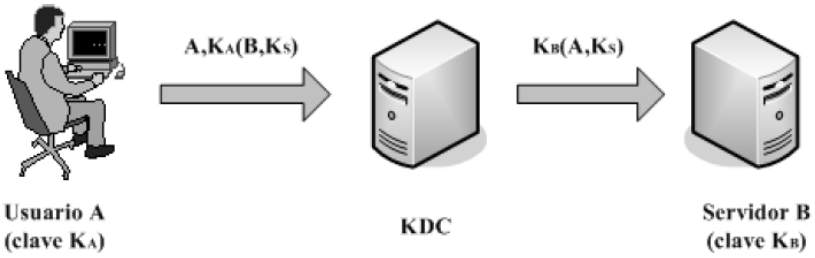

Figura 9: Establecimiento de una clave de sesión mediante un servidor KDC

La organización puede utilizar un Centro de Confianza (opción 4) para facilitar la distribución de claves en una red. Si se emplean algoritmos simétricos, el centro de confianza se denomina Servidor KDC (del idioma inglés, Key Distribution Center).

En este caso cada usuario mantiene una sola clave secreta compartida con el KDC, que se emplea para el proceso de autenticación. Este servidor KDC interviene en la administración de las claves de sesión entre los distintos usuarios (personas, servidores o dispositivos) de la red. No obstante, de este modo también puede descifrarse todos los mensajes de los usuarios, por lo que la seguridad debe ser extremada.

Para el establecimiento de una clave de sesión KS se sigue un protocolo como el siguiente:

• El usuario A y el servidor B poseen sus respectivas claves secretas KA y KB, que son reconocidas únicamente por el servidor KDC.

• El usuario A genera una clave de sesión KS por algún procedimiento previamente determinado, enviando a continuación al servidor KDC su identidad (A) y un mensaje encriptado con su clave secreta KA que contiene el identificador del servidor B con el que desea comunicarse y la clave de sesión KS.

• El servidor KDC, a su vez, envía al servidor B un mensaje encriptado con la clave secreta KB, en el que se incluye el identificador de A y la clave de sesión KS, que ha generado dicho usuario.

• De este modo, A y B pueden intercambiarse de forma segura una clave de sesión KS, empleando algoritmos de encriptación simétricos.

El enfoque de distribución automatizada de claves proporciona las características de flexibilidad y dinamismo necesarias para permitir que una serie de usuarios de terminal accedan a una serie de *hosts* y para que los *hosts* intercambien datos entre sí.

Otro enfoque para la distribución de clave utiliza el cifrado de clave pública, que se trata en la sección referida a cifrado asimétrico.

7. Cifrado asimétrico

Se basan en problemas numéricos muy complejos (como la factorización de números primos o el cálculo de logaritmos discretos). En estos sistemas se utilizan dos claves distintas: una para realizar la encriptación y otra para el proceso de desencriptación; por este motivo, reciben el nombre de asimétricos.

Un esquema de cifrado de clave pública tiene seis componentes:

• **Texto claro** consiste en el mensaje o los datos legibles que se introducen en el algoritmo como entrada.

• **Algoritmo de cifrados** el algoritmo de cifrado realiza diferentes transformaciones en el texto claro.

• **Clave pública y privada:** es una pareja de claves que han sido seleccionadas, de las cuales una se usa para el cifrado y la otra para el descifrado. Las transformaciones exactas llevadas a cabo por el algoritmo de cifrado dependen de la clave pública o privada que se proporciona como entrada.

• **Texto cifrado:** es el mensaje desordenado producido como salida. Depende del texto claro y de la clave. Para un mensaje dado, dos claves diferentes producirán dos textos cifrados diferentes.

• **Algoritmo de descifrado:** este algoritmo acepta el texto cifrado y la clave correspondiente y produce el texto claro original.

Como los nombres sugieren, la clave pública de dicha pareja de claves se hace pública para que otros la usen, mientras que la clave privada sólo es conocida por su propietario. Un algoritmo criptográfico de clave pública con propósito general se basa en una clave para el cifrado y otra diferente, aunque relacionada, para el descifrado. En este enfoque, todos los participantes tienen acceso a las claves públicas, y las claves privadas las genera cada participante de forma local y, por lo tanto, nunca necesitan ser distribuidas.

La clave empleada en el cifrado convencional se denomina comúnmente **clave secreta.** Las dos claves empleadas para el cifrado de clave pública se denominan **clave pública** y **clave privada.** Invariablemente, la clave privada se mantiene en secreto, pero en vez de llamarse clave secreta se llama clave privada para evitar confusiones con el cifrado convencional.

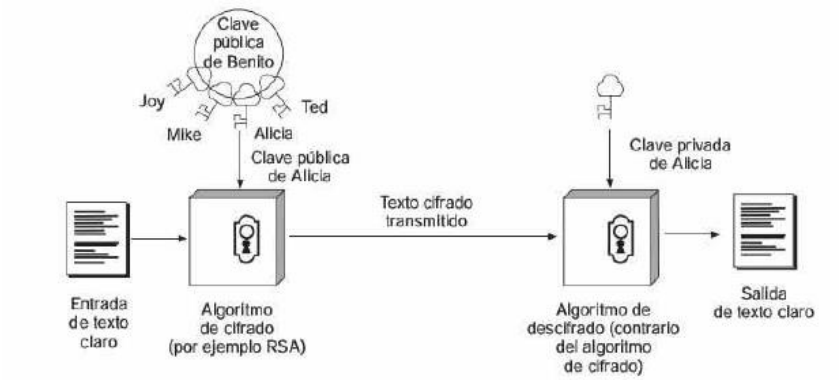

Figura 10: Encriptación con clave asimétrica

El criptosistema que se muestra en la Figura 10 depende de un algoritmo criptográfico basado en dos claves relacionadas. Diffie y Hellman postularon este sistema sin demostrar que tales algoritmos existen. Sin embargo, sí especificaron las condiciones que deben cumplir [DIFF76]:

1. Desde el punto de vista computacional, para una parte B es fácil generar una pareja de claves (clave pública K^{+}_{B}, clave privada K^{-}_{B}).

2. En términos computacionales, para un emisor A que conozca la clave pública y el mensaje que ha de cifrarse, *M,* es fácil generar el texto cifrado correspondiente: $C = E_{K+b}(M)$
3. En términos computacionales, para un receptor B es fácil descifrar el texto cifrado resultante usando la clave privada para recuperar el mensaje original:
   $M = D_{K-B}(C) = D_{K-B}[E_{K+B}(M)]$
4. Desde el punto de vista computacional, es imposible que un oponente, conociendo la clave pública, K^{+}_{B} determine la clave privada K^{-}_{B}
5. Desde el punto de vista computacional, es imposible que un oponente, conociendo la clave pública, K^{+}_{B} y un texto cifrado, *C,* recupere el mensaje original M.

Podemos añadir un sexto requisito que, aunque útil, no es necesario para todas las aplicaciones de clave pública y, por lo tanto, sólo es exigible si se desea implementar alguna de esas aplicaciones extras:

6. Cualquiera de las dos claves relacionadas puede usarse para el cifrado, y la otra para el descifrado: $M = D_{K-B}[E_{K+B}(M)] = D_{K+B}[E_{K-B}(M)]$

Conceptualmente, la utilización de un sistema de criptografía de clave pública es muy simple. Suponga que Alice quiere comunicarse con Bob. Como se muestra en la Figura 10, en lugar de que Bob y Alice compartan una única clave secreta (como es el caso en los sistemas de clave simétrica), Bob (el receptor de los mensajes de Alice) dispone en su lugar de dos claves: una clave pública que está disponible para todo el mundo (incluyendo a Trudy el intruso) y una clave privada que sólo Bob conoce. Utilizaremos la notación K^{+}_{B} y K^{-}_{B} para hacer referencia a las claves pública y privada de Bob, respectivamente. Para poder comunicarse con Bob, Alice consulta primero la clave pública de Bob y luego cifra su mensaje, m, destinado a Bob utilizando esa clave pública de Bob y un algoritmo de cifrado conocido (por ejemplo, un algoritmo estandarizado); es decir, Alice calcula $K^{+}_{B}(m)$. Bob recibe el mensaje cifrado de Alice y utiliza su clave privada y un algoritmo de descifrado conocido (por ejemplo, un algoritmo estandarizado) para descifrar el mensaje cifrado de Alice. Es decir, Bob calcula $K^{-}_{B}(K^{+}_{B}(m))$. De esta manera, Alice puede utilizar la clave de Bob que está públicamente disponible con el fin de enviar un mensaje secreto a Bob, sin que ninguno de los dos tenga que distribuir ninguna clave secreta.

Como veremos enseguida, podemos intercambiar el papel de la clave pública y la clave privada y obtener el mismo resultado; es decir, $K^{-}_{B}(K^{+}_{B}(m)) = K^{+}_{B}(K^{-}_{B}(m)) = m$. La utilización de la criptografía de clave pública es por tanto conceptualmente muy simple. Pero puede que al lector le surjan inmediatamente dos preguntas. Un primer posible problema es que, aunque un intruso que intercepte el mensaje de cifrado de Alice solo obtendrá datos sin sentido, ese intruso conoce tanto la clave (la clave pública de Bob, que está disponible para que todo el mundo la vea), como el algoritmo que Alice ha utilizado para el cifrado. Trudy podría entonces montar un ataque de texto claro conocido, utilizando ese algoritmo de cifrado estandarizado y la clave de cifrado de Bob, públicamente disponible, para codificar cualquier mensaje que desee. Trudy también podría intentar, por ejemplo, codificar mensajes, o partes de mensajes, que piense que Alice podría enviar, con el fin de suplantarla. Obviamente, para que la criptografía de clave pública pueda funcionar, la selección de claves y el cifrado/descifrado deben hacerse de forma tal que sea imposible (o al menos lo suficientemente difícil como para ser prácticamente imposible) para un intruso determinar la clave privada de Bob o descifrar o adivinar de alguna otra manera el mensaje que Alice le ha enviado a Bob. El segundo problema potencial es que, dado que la clave de cifrado de Bob es pública, cualquiera puede enviar un mensaje cifrado a Bob, incluyendo Alice o alguien que se haga pasar por Alice. Sin embargo, en el caso de una

única clave secreta compartida, este ya no será el caso puesto que nadie puede enviar un mensaje cifrado a Bob utilizando la clave públicamente disponible de este. Es necesaria una firma digital, tema que estudiaremos más adelante, para vincular a un emisor con un mensaje.

Aunque pueden existir muchos algoritmos que se correspondan con la descripción realizada, el algoritmo RSA (llamado así por sus inventores, Ron Rivest, Adi Shamir y Leonard Adleman) se ha convertido casi en sinónimo de la criptografía de clave pública.

8. Integridad de mensajes y autenticación de puntos terminales

En la sección anterior hemos visto como puede utilizarse el cifrado para proporcionar confidencialidad a dos entidades que desean comunicarse. En esta sección vamos a volver nuestra atención al tema criptográfico, igualmente importante, de proporcionar integridad a los mensajes (técnica también conocida como autenticación de mensajes). Junto con la integridad de los mensajes, analizaremos dos temas relacionados en esta sección: las firmas digitales y la autenticación de los puntos terminales. Vamos a definir el problema de la integridad de los mensajes utilizando una vez más a Alice y a Bob. Suponga que Bob recibe un mensaje (que puede estar cifrado o puede ser texto en claro) y que él cree que este mensaje fue enviado por Alice. Para autenticar el mensaje, Bob tiene que verificar que:

1. El origen del mensaje es efectivamente Alice.

2. El mensaje no ha sido alterado mientras viajaba hasta Bob.

Veremos en las secciones siguientes que este problema de la integridad de los mensajes es una preocupación crítica en prácticamente todos los protocolos de red seguros. Como ejemplo específico, considere una red de computadoras en la que se está empleando un algoritmo de enrutamiento de estado del enlace (como por ejemplo OSPF) para determinar las rutas entre cada pareja de routers de la red. En un algoritmo de estado del enlace, cada router necesita multidifundir un mensaje de estado del enlace a todos los restantes routers de la red. El mensaje de estado del enlace de un router incluye una lista de sus vecinos directamente conectados, junto con los costes directos a esos vecinos. Una vez que un router recibe mensajes de estado de enlace de todos los demás routers puede crear un mapa completo de la red, ejecutar su algoritmo Dijkstra y configurar su tabla de reenvío. Un ataque relativamente sencillo contra el algoritmo de enrutamiento consiste en que Trudy distribuya mensajes falsos de estado del enlace con información incorrecta acerca del estado de los enlaces. Debido a la necesidad de integridad de los mensajes, cuando el router B recibe un mensaje de estado del enlace procedente del router A debe verificar que efectivamente el router A ha creado dicho mensaje y, además, que nadie lo ha alterado mientras que el mensaje se encontraba en tránsito. En esta sección vamos a describir una popular técnica de integridad de mensajes que se utiliza en muchos protocolos de red seguros. Pero, antes de eso, tenemos que tratar otro tema importante dentro del campo de la criptografía: las funciones hash criptográficas.

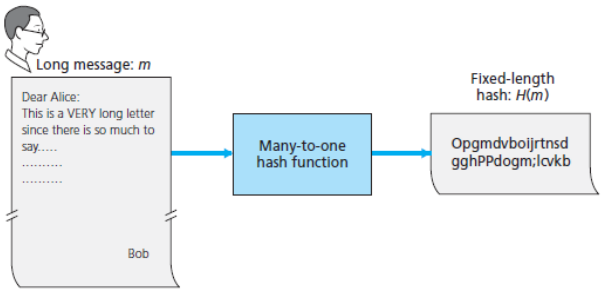

Figura 11: Funciones Hash

Funciones hash criptográficas

Como se muestra en la Figura 11, una función hash toma una entrada, *m, y* calcula una cadena de tamaño fijo *H(m)* conocida con el nombre de hash. La suma de comprobación de Internet y los códigos CRC cumplen con esta definición. Además, una función hash criptográfica necesita exhibir la siguiente propiedad adicional:

- Es computacionalmente impracticable encontrar dos mensajes distintos *x* e *y* tales que *H(x)=H(y).*

De una manera informal, podríamos decir que esta propiedad significa que es computacionalmente impracticable que un intruso sustituya un mensaje protegido mediante la función hash por otro mensaje diferente. Es decir, si (*m*, *H(m))* son el mensaje y el valor hash de dicho mensaje creado por el emisor, entonces un intruso no puede generar el contenido de otro mensaje, *y,* que tenga el mismo valor de hash que el mensaje original. Vamos a verificar que una suma de comprobación simple, como la suma de comprobación de Internet, nos proporcionaría una función hash criptográfica bastante poco segura. En lugar de realizar la aritmética en complemento a 1 (como en la suma de comprobación de Internet), calculemos una suma de comprobación tratando cada caracter como un byte y sumando los bytes en fragmentos de 4 bytes cada vez. Suponga que Bob debe a Alice 100,99 euros y que le envía un mensaje de confirmación que es la cadena de texto `"DEBO100.99BENITO."` La representación ASCII (en notación hexadecimal) de estas letras sería `44, 45, 42, 4F,31, 30, 30, 2E, 39, 33, 42, 45, 4E, 49, 54, 4F.` La Figura 12 (parte superior) muestra que la suma de comprobación de 4 bytes para este mensaje es FC F8 09 11. Un mensaje ligeramente distinto (y mucho más costoso para Bob) es el que se muestra en la parte inferior de esta misma figura. Los mensajes `"DEBO100.99BENITO"` y `"DEBO900.19BENITO"` tienen la *misma* suma de comprobación. Por tanto, este sencillo algoritmo de suma de comprobación violaría el requisito que antes hemos mencionado. Dados los datos originales, es muy sencillo encontrar otro conjunto de datos con la misma suma de comprobación. Obviamente, para propósitos de seguridad necesitaremos una función hash bastante más potente que una mera suma de comprobación. El algoritmo hash MD5 de Ron Rivest [RFC 1321] se utiliza ampliamente hoy día. Este algoritmo calcula un valor hash de 128 bits mediante un proceso en cuatro pasos. Consulte [RFC 1321] para ver una descripción de MD5 (incluyendo una implementación con código fuente en C). El segundo algoritmo principal de hash que se utiliza hoy día es el denominado algoritmo de hash seguro (SHA-1, *Secure Hash Algorithm)* [FIPS 1995], Este algoritmo está basado en una serie de principios similares a los utilizados en el diseño de MD4 [RFC 1320], el predecesor de MD5. SHA-1 produce un resumen del mensaje *(message digest)* de 160 bits. Esa mayor longitud de salida hace que SHA-1 sea más seguro, sin embargo, SHA-1 está siendo desplazado por SHA-256 que produce una salida de 256 bits.

Message	ASCII Representation				
I O U 1	49	4F	55	31	
0 0 . 9	30	30	2E	39	
9 B O B	39	42	4F	42	
	B2	C1	D2	AC	Checksum

Message	ASCII Representation				
I O U 9	49	4F	55	39	
0 0 . 1	30	30	2E	31	
9 B O B	39	42	4F	42	
	B2	C1	D2	AC	Checksum

Figura 12: El mensaje inicial y el mensaje fraudulento

Código de autenticación del mensaje

Volvamos ahora al problema de la integridad de los mensajes. Ahora que sabemos que son las funciones hash, veamos una primera aproximación de cómo podríamos garantizar la integridad de los mensajes:

1. Alice crea el mensaje *m* y calcula el valor hash *H(m)* (por ejemplo, con SHA-1).

2. Alice añade a continuación *H(m)* al mensaje *m,* creando un mensaje ampliado *(m, H(m)),* el cual envía a Bob.

3. Bob recibe el mensaje ampliado (*m*, *h*) y calcula *H(m).* Si *H(m)* = *h,* Bob concluye que todo está correcto.

Este enfoque tiene un fallo fundamental. El intruso Trudy puede crear un mensaje ficticio *m'* en el que dijera que es Alice, calcular *H(m')* y enviar a Bob (m', *H(m')).* Cuando Bob recibiera el mensaje, todas las comprobaciones del paso 3 serían correctas, por lo que Bob no sería consciente de que se ha producido un engaño. Para garantizar la integridad de los mensajes, además de utilizar funciones hash criptográficas Alice y Bob necesitan un secreto compartido s. Este secreto compartido, que no es más que una cadena de bits, se denomina clave de autenticación. Utilizando este secreto compartido puede garantizarse de la forma siguiente la integridad de los mensajes:

1. Alice crea el mensaje m, concatena *s* con *m* para crear *m + s,* y calcula el valor hash *H(m +* s) (por ejemplo, con SHA-1). *H(m* + s) se denomina código de autenticación de mensajes (MAC, *Message Auihentication Code).*

2. Alice añade entonces el código MAC al mensaje *m*, creando un mensaje ampliado *(m, H(m* + *s*)), y lo envía a Bob.

3. Bob recibe un mensaje ampliado (*m, h*) y, conociendo *s,* calcula el valor MAC *H(m + s).* Si *H(m + s) = h,* Bob concluye que todo está correcto.

En la Figura 13 se muestra un resumen de este procedimiento. El lector debe fijarse en que las siglas MAC aquí (que corresponden a "*Message Authentication Code")* no tienen nada que ver con las siglas MAC utilizadas en los protocolos de la capa de enlace (que corresponden a *"Medium Access Control").* Una característica muy conveniente de los valores MAC es que no se necesita ningún algoritmo de cifrado. De hecho, en muchas aplicaciones, incluyendo el algoritmo de enrutamiento de estado del enlace descrito anteriormente, las entidades que se están comunicando sólo se preocupan de la integridad de los mensajes, mientras que la confidencialidad de estos no les importa. Utilizando

un código MAC, las entidades pueden autenticar los mensajes que se intercambian sin tener que incluir complejos algoritmos de cifrado en el proceso de garantía de la integridad.

Como cabría esperar, a lo largo de los años se han propuesto diversos estándares para los valores MAC. El estándar más popular hoy día es HMAC. En la práctica, HMAC hace pasar los datos y la clave de autenticación a través de la función hash dos veces [Kaufman 1995; RFC 2104].

Sigue quedando pendiente una cuestión importante. ¿Cómo distribuimos la clave secreta de autenticación a las distintas entidades que tienen que comunicarse? Por ejemplo, en el algoritmo de enrutamiento de estado del enlace necesitaremos distribuir de alguna manera la clave de autenticación a cada uno de los routers del sistema autónomo. (Observe que todos los routers pueden utilizar la misma clave de autenticación.) Un administrador de red podría llevar a cabo esa distribución visitando físicamente cada uno de los routers. O bien, si el administrador de la red es demasiado perezoso y si cada router tiene su propia clave pública, el administrador puede distribuir la clave de autenticación a cualquiera de los routers cifrándola con la clave pública del router y luego enviando la clave cifrada hasta el router a través de la red.

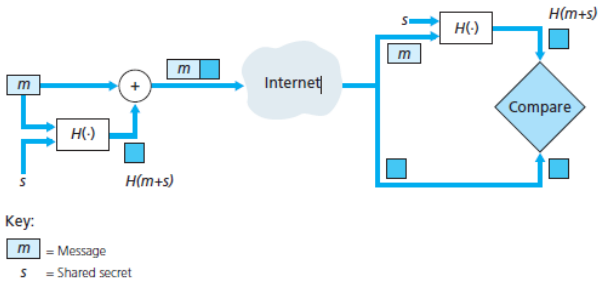

Figura 13: Código de autenticación de mensaje (MAC)

Firma digital

El cifrado de clave pública se puede utilizar de otra forma, como se muestra en la Figura 14. Suponga que Bob quiere enviar un mensaje a Alice y, aunque no es importante que el mensaje se mantenga secreto, quiere que Alice tenga la certeza de que el mensaje proviene efectivamente de él. Suponga además que se cumple la condición antes mencionada: *Cualquiera de las dos claves relacionadas puede usarse para el cifrado, y la otra para el descifrado: M = DK-B [EK+B (M)] = DK+B [EK-B (M)].*

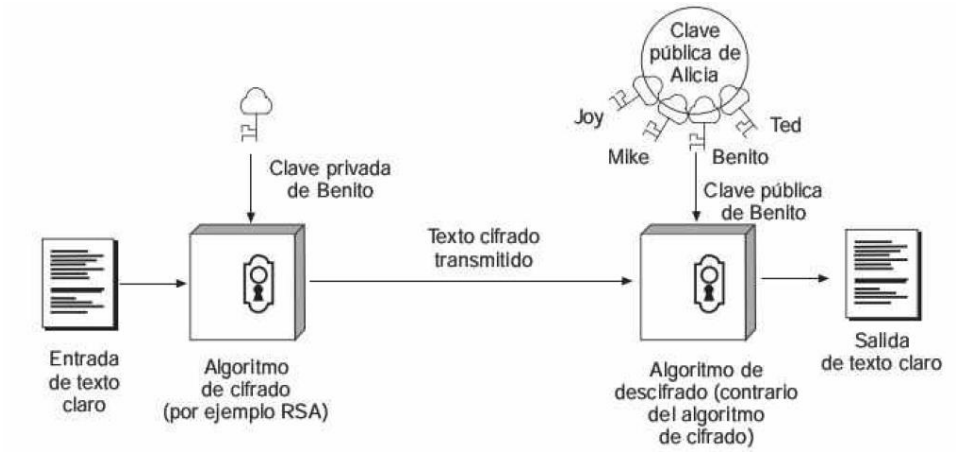

Figura 14: Autentificación de mensaje y no repudio

Piense en el número de veces que ha firmado con su nombre en un papel durante la última semana. Todos estamos acostumbrados a firmar cheques, recibos de tarjetas de crédito, documentos legales y cartas. Nuestra firma atestigua el hecho de que nosotros (y no cualquier otra persona) hemos aceptado y/o acordado el contenido de un documento. En un mundo digital a menudo surge la necesidad de indicar el propietario o creador de un documento o de explicitar nuestro acuerdo con el contenido de un documento. Una **firma digital** es una técnica criptográfica que permite conseguir estos objetivos en el mundo digital.

Al igual que ocurre con las firmas manuscritas, las firmas digitales deben realizarse de forma que sean verificables y no falsificables. Es decir, debe ser posible demostrar que un documento firmado por una persona ha sido, de hecho, firmado por esa persona (la firma tiene que ser verificable) y que *sólo* esa persona podría haber firmado el documento (la firma no puede ser falsificada).

Consideremos ahora como podríamos diseñar un esquema de firma digital. Observe que, cuando Bob firma un mensaje, debe poner algo en el mensaje que sea distintivo de él. Bob podría pensar en asociar un valor MAC para la firma, en cuyo caso crearía el valor MAC añadiendo su clave (que permite distinguirle del resto de las personas) al mensaje y luego aplicando la función hash. Pero, para que Alice pudiera verificar esa firma, también debería disponer de una copia de la clave, en cuyo caso la clave dejaría de ser distintiva de Bob. Por tanto, los códigos MAC no nos permiten conseguir nuestro objetivo en este caso.

Recuerde que, con la criptografía de clave pública, Bob dispone de sendas claves pública y privada, siendo la pareja formada por esas claves distintiva de Bob. Por tanto, la criptografía de clave pública es un candidato excelente para poder proporcionar un mecanismo de firma digital. Veamos ahora cómo se estructura dicho mecanismo

Suponga que Bob desea firmar digitalmente un documento, *m.* Podemos pensar en el documento como si fuera un archivo o un mensaje que Bob va a firmar y enviar. Como se muestra en la Figura 15, para firmar este documento, Bob utiliza simplemente *su* clave privada, K_B^-, para calcular $K_B^-(m)$. De entrada, puede parecer extraño que Bob utilice su clave privada (que, como vimos, se utiliza para descifrar un mensaje que haya sido cifrado con su clave pública) para firmar un documento. Pero recuerde que el cifrado y el descifrado no son otra cosa que operaciones matemáticas (consistentes en elevar a la potencia *e o d* en RSA) y recuerde también que el objetivo de Bob no es cifrar u ocultar el

contenido del documento, sino sólo firmarlo de una manera que sea verificable y no falsificable. La firma digital del documento realizada por Bob es $K_B^-(m)$.

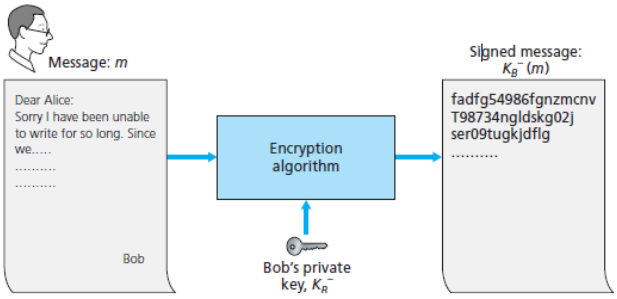

Figura 15: Creación de una firma digital para un documento

¿Cumple la firma digital $K_B^-(m)$ nuestros requisitos de que sea verificable y no *falsificable?* Suponga que Alice dispone de *m* y $K_B^-(m)$. Ella quiere demostrar ante un tribunal (a consecuencia de algún tipo de litigio) que Bob es quien ha firmado el documento y que es la única persona que podría haberlo firmado. Alice toma la clave pública de Bob, K_B^+, y se la aplica a la firma digital, $K_B^-(m)$, asociada con el documento, *m*. Es decir, calcula $K_B^+(K_B^-(m))$, y voila: con un gesto dramático obtiene *m,* que se corresponde exactamente con el documento original. Alice argumenta a continuación que sólo Bob podría haber firmado el documento por las siguientes razones:

- Quienquiera que haya firmado el documento tiene que haber utilizado la clave privada de Bob, K_B^-, a la hora de calcular la firma $K_B^-(m)$ para que $K_B^+(K_B^-(m)) = m$.
- La única persona que puede conocer la clave privada, K_B^-, es el propio Bob. Recuerde que el conocimiento de la clave pública, K_B^+, no sirve de nada a la hora de determinar la clave privada, K_B^-. Por tanto, la única persona que podría conocer K_B^- es aquella que haya generado la pareja de claves, (K_B^-, K_B^+), es decir, Bob. (Observe que aquí se está suponiendo, sin embargo, que Bob no ha proporcionado su clave privada K_B^- a nadie y que nadie le ha robado K_B^-.)

También es importante observar que si el documento original, *m,* se modificara de algún modo para obtener una forma alternativa, *m',* la firma que Bob generara para *m* no sería válida para *m',* ya que $K_B^+(K_B^-(m))$ no es igual a *m'*. Por tanto, podemos concluir que las firmas digitales también proporcionan un mecanismo de integridad de los mensajes, permitiendo al receptor verificar que el mensaje no ha sido alterado, además de verificar el origen del mismo.

En síntesis, nadie más tiene la clave privada de Bob y, por tanto, nadie más ha podido crear el texto cifrado que pudo ser descifrado con su clave pública. Además, es imposible alterar el mensaje sin acceder a la clave privada de Bob, por lo que el mensaje está autenticado en términos de origen e integridad de los datos.

Uno de los problemas con la firma de datos mediante mecanismos de cifrado es que el cifrado y el descifrado son computacionalmente muy caros. Dada la cantidad adicional de procesamiento que el cifrado y el descifrado exigen, el firmar los datos vía cifrándolos/descifrándolos completamente puede ser como matar moscas a cañonazos. **Una técnica más eficiente consiste en introducir funciones hash en el mecanismo de firma digital.** Recuerde que un algoritmo hash toma un mensaje, *m,* de longitud arbitraria y calcula una especie de "huella digital" de longitud fija para el mensaje, que

designamos mediante *H(m).* Utilizando una función hash, Bob firma el valor hash de un mensaje en lugar de firmar el propio mensaje, es decir, Bob calcula $K^-_B(H(m))$. Puesto que *H(m)* es, generalmente, mucho más pequeño que el mensaje original *m,* la capacidad de proceso necesario para generar la firma digital se reduce sustancialmente.

En el contexto del envío de un mensaje a Alice por parte de Bob, la Figura 16 proporciona un resumen del procedimiento operativo requerido para crear una firma digital. Bob hace pasar su mensaje original, de gran longitud, a través de una función hash. A continuación, firma digitalmente el valor hash resultante utilizando para ello su clave privada. Después, le envía a Alice el mensaje original (como texto en claro) junto con el resumen del mensaje digitalmente firmado (al que a partir de ahora denominaremos firma digital). La Figura 17 muestra el procedimiento en Alice para verificar la firma digital. Alice aplica la clave pública del emisor al mensaje para obtener un valor hash. Asimismo, aplica la función hash al mensaje recibido como texto en claro, para obtener un segundo valor hash. Si los dos valores coinciden, entonces Alice puede estar segura acerca de la integridad y del autor del mensaje.

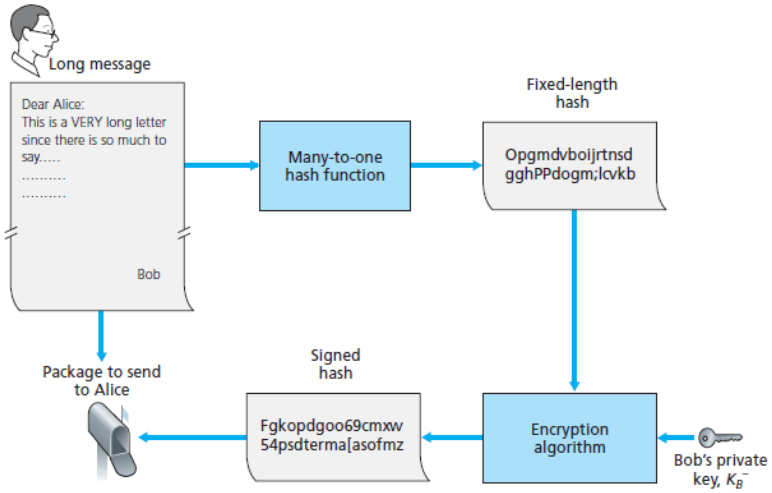

Figura 16: Transmisión de un mensaje firmado digitalmente

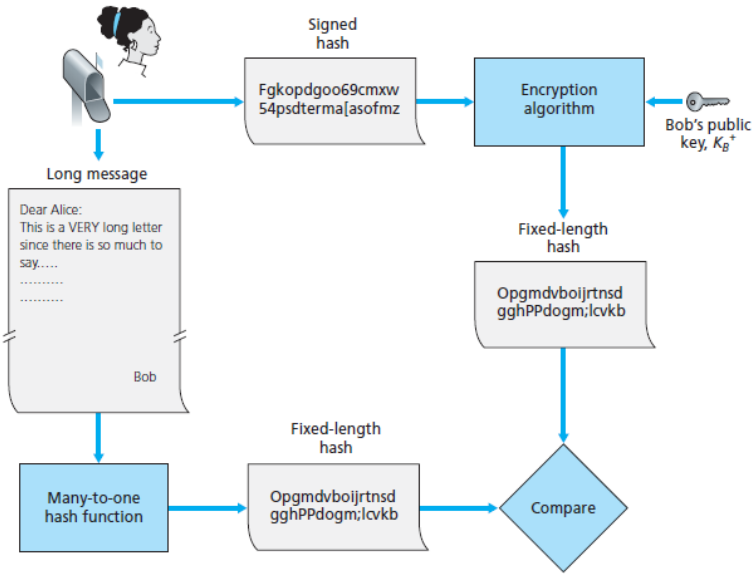

Figura 17: Verificación de un mensaje firmado

Antes de seguir adelante, vamos a comparar brevemente las firmas digitales con los códigos MAC, dado que existen paralelismos entre ambos sistemas, pero también existen varias diferencias sutiles e importantes. Tanto las firmas digitales como los códigos MAC comienzan con un mensaje (o un documento). Para obtener un valor MAC de ese mensaje, añadimos al mensaje una clave de autenticación y luego calculamos el valor hash del resultado. Observe que, a la hora de crear el valor MAC, no hay involucrado ningún mecanismo de cifrado de clave pública ni de clave simétrica. Para crear una firma digital, primero calculamos el valor hash del mensaje y luego ciframos el mensaje con nuestra clave privada (utilizando criptografía de clave pública). Por tanto, la firma digital es una técnica "más pesada", dado que requiere una Infraestructura de clave pública (PKI, *Public Key Infrastructure)* subyacente, en la que existan autoridades de certificación, como las que más adelante se describen. OSPF utiliza valores MAC para garantizar la integridad de los mensajes. Veremos que los valores MAC también se emplean en varios protocolos de seguridad populares de las capas de transporte y de red.

Síntesis del uso de critpografía asimétrica para confidencialidad, integridad y no repudio

Reuniendo los conceptos ya vistos, la firma electrónica de un mensaje o transacción permite garantizar la integridad, la autenticación y la no repudiación de un sistema de información basado en TICs. Para su obtención se sigue un esquema bastante sencillo: el creador de un mensaje debe encriptar la "huella digital" del mensaje con su clave privada y enviarla al destinatario acompañando al mensaje encriptado. La encriptación asimétrica se aplica a la "huella digital" del mensaje y no sobre el propio mensaje, debido al elevado costo computacional que supone la encriptación de todo el mensaje, alternativa que resulta mucho más lenta y compleja.

La Figura 18 muestra el procedimiento seguido por el Usuario A para enviar un mensaje encriptado a otro usuario B acompañado de la correspondiente firma electrónica.

Una vez recibido el mensaje encriptado por A, el usuario B realiza los siguientes pasos para comprobar la autenticidad y la integridad del mensaje:

1. Recupera el mensaje original desencriptando el texto cifrado con su clave privada. Como sólo él conoce esta clave, se garantiza la confidencialidad en la red.

2. Aplica el algoritmo de compendio (algoritmo hash) para generar la "huella digital" del mensaje que acaba de recibir.

3. Utiliza la clave pública de A para desencriptar la "huella digital" del mensaje original. Conviene recordar que esta "huella digital" había sido encriptada por el usuario A con su clave privada (constituía la firma electrónica de A sobre el mensaje original).

4. Compara la "huella digital" desencriptada con la que acaba de generar a partir del mensaje recibido y, si ambas coinciden, podrá estar seguro de que el mensaje es auténtico y se ha respectado su integridad.

En definitiva, con el esquema propuesto basado en un sistema criptográfico y la firma electrónica se consigue garantizar la confidencialidad, la integridad y la autenticación de los mensajes transmitidos.

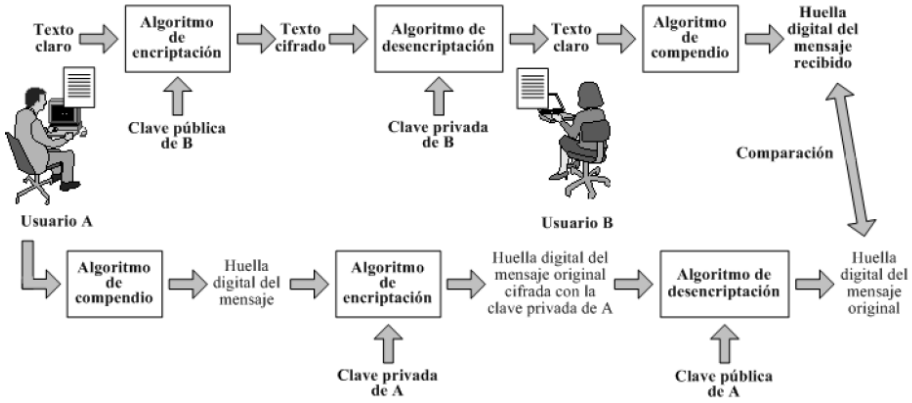

Figura 18: Utilización de la firma electrónica o digital con aseguramiento de confidencialidad

9. Gestión de claves públicas

Sin embargo, hay un problema que hemos pasado por alto demasiado rápido: siAlice y Bob no se conocen entre sí, ¿cómo obtiene cada uno la clave pública del otro para iniciar el proceso de comunicación? La solución más obvia -colocar su clave pública en su sitio web- no funciona por la siguiente razón: Suponga que Alice quiere buscar la clave pública de Bob en el sitio web de él. ¿Cómo lo hace? Comienza tecleando el URL de Bob. A continuación, su navegador busca la dirección DNS de la página de inicio de Bob y le envía una solicitud GET, como se muestra en la Figura 19. Por desgracia, Trudy intercepta la solicitud y responde con una página de inicio falsa, quizás una copia de la de Bob excepto porque reemplaza la clave pública de Bob con la de Trudy. Cuando Alice encripta su primer mensaje mediante E_T , Trudy lo desencripta, lo lee, lo vuelve a encriptar con la clave pública de Bob y lo envía a éste, quien no tiene la menor idea de que Trudy está leyendo los mensajes que le llegan. Peor aún, Trudy puede modificar los mensajes antes de volverlos a encriptar para Bob. Es evidente que se necesita un mecanismo para asegurar que las claves públicas se puedan intercambiar de manera segura.

Certificados

Como un primer intento por distribuir claves públicas de manera segura, podemos imaginar un centro de distribución de claves KDC disponible en línea las 24 horas del día, que proporciona claves públicas bajo demanda. Uno de los muchos problemas con esta solución es que no es escalable, y el centro de distribución de claves podría convertirse rápidamente en un “cuello de botella”. Además, si alguna vez fallara, la seguridad en Internet se paralizaría por completo.

Por estas razones se desarrolló una solución diferente, una que no requiere que el centro de distribución de claves esté en línea todo el tiempo. De hecho, ni siquiera tiene que estar en línea. En su lugar, lo que hace es certificar las claves públicas que pertenecen a las personas, empresas y otras organizaciones. Ahora, a una organización que certifica claves públicas se le conoce como CA (Autoridad de Certificación, del inglés *Certification Authority).*

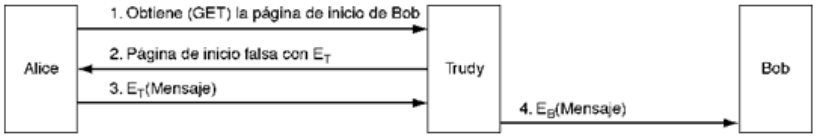

Figura 19: Una forma en la que Trudy puede destruir la encriptación de clave pública

Como un ejemplo, suponga que Bob desea permitir que Alice y otras personas que no conoce se comuniquen con él de manera segura. Él puede ir con la CA con su clave pública, junto con su pasaporte o licencia de conducir para pedir su certificación. A continuación, la CA emite un certificado similar al que se muestra en la Figura 20 y firma su hash SHA-1 con la clave privada de la CA (un mecanismo más seguro utiliza SHA-256). Luego Bob paga la cuota de la CA y obtiene el certificado con su hash firmado.

Figura 20: Un posible certificado y su hash firmado

El trabajo fundamental de un certificado es enlazar una clave pública con el nombre de un personaje principal (individuo, empresa, etc.). Los certificados no son secretos ni están protegidos. Por ejemplo, Bob podría tomar la decisión colocar su nuevo certificado en su sitio web, con un vínculo en la página de inicio que diga “Haga clic aquí para obtener mi certificado de clave pública”. El clic resultante podría regresar el certificado y el bloque de la firma (el hash SHA-1 firmado del certificado).

Ahora vayamos otra vez al escenario que se muestra en la Figura 19. Cuando Trudy intercepta la solicitud que Alice realiza para obtener la página de inicio de Bob, ¿qué puede hacer Trudy? Puede poner su propio certificado y bloque de firma en la página falsificada, pero cuando Alice lea el certificado, verá inmediatamente que no está hablando con Bob porque el nombre de éste no se encuentra en dicho certificado. Trudy puede modificar sobre la marcha la página de inicio de Bob y reemplazar la clave pública de Bob con la suya. Sin embargo, cuando Alice ejecute el algoritmo SHA-1 sobre el certificado, obtendrá un hash que no corresponde con el que obtiene al aplicar la clave pública reconocida de la CA al bloque de firma. Puesto que Trudy no tiene la clave privada de la CA, no tiene forma de generar un bloque de firma que contenga el hash de la página web modificada con su clave pública en él. De esta manera, Alice puede estar segura de que tiene la clave pública de Bob y no la de Trudy o la de alguien más. Y, como prometimos, este esquema no requiere que la CA esté en línea para la verificación, por lo que se elimina un potencial “cuello de botella”.

Mientras que la función estándar de un certificado es enlazar una clave pública a un personaje principal, un certificado también se puede utilizar para enlazar una clave pública a un atributo. Por ejemplo, un certificado podría decir: “Esta clave pública pertenece a alguien mayor de 18 años”. Podría utilizarse para probar que el dueño de la clave privada no es una persona menor de edad y se le permita acceder a cierto material no apto para niños, entre otras cosas, pero sin revelar la identidad del dueño. Por lo general, la persona que tiene el certificado podría enviarlo al sitio web, al personaje principal o al proceso que se preocupa por la edad. El sitio, personaje principal o proceso podría generar a continuación un número aleatorio y encriptarlo con la clave pública del certificado. Si el dueño pudiera desencriptarlo y regresarlo, ésa sería una prueba de que el dueño tenía el atributo establecido en el certificado. De manera alternativa, el número aleatorio podría utilizarse para generar una clave de sesión para la conversación resultante.

X.509

Si todas las personas que quisieran algo firmado fueran a la CA con un tipo diferente de certificado, la administración de todos los distintos formatos pronto se volvería un problema. Para resolverlo se ha diseñado un estándar para certificados, aprobado por la ITU. Dicho estándar se conoce como X.509 y se utiliza mucho en Internet. Ha pasado por tres versiones desde su estandarización inicial en 1988. Aquí analizaremos la versión V3.

El X.509 ha recibido una enorme influencia del mundo de OSI, y ha tomado prestadas algunas de sus peores características (por ejemplo, la asignación de nombres y la codificación). Lo sorprendente es que la IETF estuvo de acuerdo con el X.509, aun cuando en casi todas las demás áreas, desde direcciones de máquinas y protocolos de transporte hasta los formatos de correo electrónico, la IETF por lo general ignoró a la OSI y trató de hacerlo bien. La versión del X.509 de la IETF se describe en el RFC 5280.

En esencia, el X.509 es una forma de describir certificados. Los campos principales en un certificado se listan en la Figura 21. Las descripciones que se establecen en esa figura deben proporcionar una idea general de lo que hacen los campos. Para información adicional, por favor consulte el estándar mismo o el RFC 2459.

Por ejemplo, si Bob trabaja en el Departamento de Préstamos del Banco Monetario, su dirección X.500 podría ser:

/C=MX/O=BancoMonetario/OU=Prestamo/CN=Bob/

donde *C* corresponde al país, *O* a la organización, *OU* a la unidad organizacional y *CN* a un nombre común. Las CA y otras entidades se nombran de forma similar. Un problema considerable con los nombres X.500 es que, si Alice está tratando de contactar a bob@bancomonetario.com y se le asigna un certificado con un nombre X.500, tal vez no sea obvio para ella que el certificado se refiera al Bob que ella busca. Por fortuna, a partir de la versión 3 se permiten los nombres DNS en lugar de los nombres X.500.

Los certificados están codificados mediante la ASN.1 (Notación de Sintaxis Abstracta 1, del inglés *Abstract Syntax Notation 1)* de la OSI, que puede considerarse como si fuera una estructura de C, pero con una notación muy peculiar y prolija. Es posible encontrar información acerca de X.509 en Ford y Baum (2000).

Campo	Significado
Versión	Qué versión de X.509.
Número de serie	Este número más el nombre de la CA identifican el certificado de manera única.
Algoritmo de firma	El algoritmo utilizado para firmar el certificado.
Emisor	El nombre X.500 de la CA.
Periodo de validez	Los tiempos inicial y final del periodo de validez.
Nombre del sujeto	La entidad cuya clave se va a certificar.
Clave pública	La clave pública del sujeto y la ID del algoritmo que la utiliza.
ID del emisor	Un ID opcional que identifica al emisor del certificado en forma única.
ID del sujeto	Un ID opcional que identifica al sujeto del certificado en forma única.
Extensiones	Se han definido muchas extensiones.
Firma	La firma del certificado (firmada por la clave privada de la CA).

Figura 21: Los campos básicos de un certificado X.509

Infraestructura de clave pública

El hecho de que una sola CA emitiera todos los certificados del mundo obviamente no funcionaría. Podría derrumbarse por la carga y también podría ser un punto central de fallas. Una posible solución sería tener múltiples autoridades CA que fueran operadas por la misma organización y que utilizaran la misma clave privada para firmar los certificados. Si bien esto podría solucionar los problemas de carga y de fallas, introduciría un nuevo problema: la fuga de claves. Si hubiera docenas de servidores esparcidos por todo el mundo, todos con la misma clave privada de la CA, la probabilidad de que esta clave fuera robada o se filtrara de algún otro modo se incrementaría de manera considerable. Puesto que la situación comprometida de esta clave arruinaría la infraestructura de la seguridad electrónica mundial, tener una sola CA central es muy peligroso.

Además, ¿qué organización podría operar la CA? Es difícil imaginar cualquier autoridad que se pudiera aceptar mundialmente como legítima y digna de confianza. En algunos países las personas insistirían en que fuera el gobierno, mientras que en otros rechazarían esta opción por completo.

Por estas razones se ha desarrollado una forma diferente para certificar claves públicas. Su nombre general es **PKI** (**Infraestructura de Clave Pública**, del inglés *Public Key Infrastructure*). En esta sección resumiremos cómo funciona en general, aunque se han generado diversas propuestas, por lo que es probable que los detalles cambien con el tiempo.

Una PKI tiene varios componentes: usuarios, autoridades de certificación (CA), certificados y directorios. Lo que la PKI hace es proporcionar una forma de estructurar estos componentes y definir estándares para los diversos documentos y protocolos. Una forma particularmente simple de PKI es una jerarquía de autoridades CA, como se muestra en la Figura 22. En este ejemplo mostramos tres niveles, pero en la práctica podrían ser menos o más. La CA de nivel superior, la RAÍZ, certifica a las autoridades CA de segundo nivel. A su vez, estas autoridades CA de segundo nivel certifican a las CA reales, las cuales emiten los certificados X.509 a organizaciones e individuos. Cuando la raíz autoriza una nueva CA, genera un certificado X.509 donde indica que ha aprobado la CA, e incluye en él la nueva clave pública de la CA, la firma y se la entrega a la CA. De manera similar, cuando una CA

aprueba una nueva CA de nivel inferior, produce y firma un certificado que indica su aprobación y que contiene la clave pública de la CA.

Nuestra PKI funciona como se muestra a continuación. Suponga que Alice necesita la clave pública de Bob para comunicarse con él, por lo que busca y encuentra un certificado que la contenga, firmado por la CA 5. Sin embargo, Alice nunca ha escuchado acerca de la CA 5. En lo que a ella respecta, la CA 5 podría ser la hija de 10 años de Bob. Podría ir con la CA 5 y decirle: "Prueba tu autenticidad". La CA 5 le responderá con el certificado que obtuvo de la CA 7, el cual contiene la clave pública de la CA 5. Una vez que tenga la clave pública de la CA 5, Alice podrá verificar que el certificado de Bob realmente fue firmado por la CA 5 y que, por lo tanto, es legal.

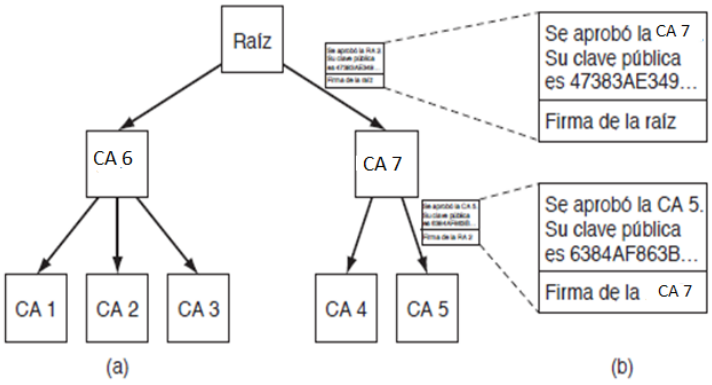

Figura 22: (a) Una PKI jerárquica. (b) Una cadena de certificados

A menos que la CA 7 sea el hijo de 12 años de Bob. Así, el siguiente paso es que Alice pida a la RA 2 que pruebe su autenticidad. La respuesta a su consulta es un certificado firmado por la raíz que contiene la clave pública de la CA 7. Ahora Alice está segura de que tiene la clave pública de Bob.

Pero ¿cómo averigua Alice la clave pública de la raíz? Magia. Se supone que todos conocen la clave pública de la raíz. Por ejemplo, su navegador podría tener integrada la clave pública de la raíz.

Bob es una persona amigable y no desea causar a Alice tanto trabajo. Él sabe que ella va a tener que verificar la CA 5 y la CA 7, por lo que, para ahorrarle algunos problemas, recolecta los dos certificados necesarios y se los proporciona junto con el de él. Ahora ella puede utilizar el conocimiento que tiene de la clave pública de la raíz para verificar el certificado de nivel superior y la clave pública contenida ahí para verificar la segunda. De esta manera, Alice no necesita contactar a nadie para realizar la verificación. Puesto que todos los certificados están firmados, Alice puede detectar con facilidad cualquier intento de alterar el contenido. En ocasiones, a una cadena de certificados que va de esta forma a la raíz se denomina cadena de confianza o ruta de certificación. La técnica se utiliza ampliamente en la práctica.

Por supuesto, aún tenemos el problema de quién va a ejecutar la raíz. La solución no es tener una sola raíz, sino tener muchas, cada una con sus propias autoridades ra y CA. De hecho, los navegadores modernos vienen precargados con claves públicas para cerca de 100 raíces, algunas veces llamadas anclas de confianza. De esta forma, es posible evitar tener una sola autoridad mundial de confianza.

Pero ahora queda el problema de cómo decide el fabricante del navegador cuáles anclas de confianza propuestas son confiables y cuáles no. Queda a criterio del usuario confiar en el fabricante

del navegador para elegir las mejores opciones y no simplemente aprobar todas las anclas de confianza, dispuesto a pagar sus cuotas de inclusión. La mayoría de los navegadores permiten que los usuarios inspeccionen las claves de la raíz (por lo general en la forma de certificados firmados por la raíz) y eliminen las que parezcan sospechosas.

Revocación

El mundo real también está lleno de certificados, como los pasaportes y las licencias de conducir. Algunas veces estos certificados pueden revocarse; por ejemplo, las licencias de conducir pueden revocarse por conducir en estado de ebriedad y por otros delitos de manejo. En el mundo digital ocurre el mismo problema: el otorgante de un certificado podría decidir revocarlo porque la persona u organización que lo posee ha abusado de él en cierta manera. También puede revocarse si la clave privada del sujeto se ha expuesto o, peor aún, si la clave privada de la CA ha sido comprometida. Por lo tanto, una PKI necesita lidiar con el problema de la revocación. La posibilidad de la revocación complica las cosas.

Un primer paso en esta dirección es hacer que cada CA emita en forma periódica una CRL (Lista de Revocación de Certificados, del inglés *Certificate Revocation List)* que proporcione los números seriales de todos los certificados que ha revocado. Puesto que los certificados contienen tiempos de expiración, la CRL sólo necesita contener los números seriales de los certificados que no han expirado todavía. Una vez que pasa el tiempo de expiración de un certificado, éste se invalida de manera automática, por lo que no hay necesidad de hacer una distinción entre los certificados que han expirado y los que fueron revocados. En ninguno de los casos se puede usar uno de esos certificados.

Por desgracia, introducir listas CRL significa que un usuario que está próximo a utilizar un certificado debe adquirir la CRL para ver si éste se revocó. Si es así, dicho certificado no debe utilizarse. Sin embargo, si el certificado no está en la lista, pudo haber sido revocado justo después de que se publicó la lista. Por lo tanto, la única manera de estar seguro realmente es preguntar a la CA. Y la siguiente vez que se utilice ese mismo certificado, se le tiene que preguntar de nuevo a la CA, puesto que pudo haber sido revocado segundos antes.

Otra complicación es que un certificado revocado puede reinstalarse nuevamente; por ejemplo, si se revocó por no pagar una cuota que ya se encuentra al corriente. Al tener que lidiar con la revocación (y quizá con la reinstalación), se elimina una de las mejores propiedades de los certificados; es decir, que pueden utilizarse sin tener que contactar a una CA.

¿Dónde deben almacenarse las listas CRL? Un buen lugar sería el mismo en el que se almacenan los certificados. Una estrategia es que una CA quite de manera activa y periódica listas CRL y hacer que los directorios las procesen con sólo eliminar los certificados revocados. Si no se utilizan directorios para almacenar certificados, las listas CRL pueden almacenarse en caché en varias ubicaciones alrededor de la red. Puesto que una CRL es por sí misma un documento firmado, si se altera, esa alteración puede detectarse con facilidad.

Si los certificados tienen tiempos de vida largos, las listas CRL también los tendrán. Por ejemplo, si las tarjetas de crédito son válidas durante 5 años, el número de revocaciones pendientes será mucho más grande que si se emitieran nuevas tarjetas cada 3 meses. Una forma estándar para tratar con grandes listas CRL es emitir algunas veces una lista maestra, pero emitir actualizaciones con más frecuencia. Hacer esto reduce el ancho de banda necesario para distribuir las listas CRL.

Para facilitar el trabajo, es posible utilizar el Protocolo de comprobación del Estado de un Certificado En línea u Online Certificate Status Protocol (OCSP), que es un método para determinar el estado de vigencia de un certificado digital X.509 usando otros medios que no sean el uso de CRL (Listas de Revocación de Certificados). Este protocolo se describe en el RFC 6960 y su estudio no forma parte del curso.

10. Capa de socket segura

Transport Layer Security (TLS) es uno de los protocolos de seguridad más importantes y ampliamente utilizados. Protege una proporción significativa de los datos que se transmiten en línea. Es más prominente se utiliza para proteger los datos que viajan entre un navegador web y un sitio web a través de HTTPS, pero también se puede usar para proteger el correo electrónico y una gran cantidad de otros protocolos.

TLS es valioso porque garantiza que la otra parte en una conexión sea quien dice ser, muestra si los datos conservan su integridad inicial y proporciona confidencialidad a través del cifrado.

TLS utiliza una variedad de algoritmos y esquemas diferentes para lograr estos propósitos. Esta sección se referirá al funcionamiento de TLS y no a los algoritmos de encriptación.

¿Qué hace TLS?

Al enviar información en línea, nos encontramos con tres problemas principales de seguridad:

1. ¿Cómo podemos saber si la persona con la que nos estamos comunicando es realmente quien dice ser?
2. ¿Cómo podemos saber que los datos no han sido alterados desde que los enviaron?
3. ¿Cómo podemos evitar que otras personas vean y accedan a los datos?

Estos problemas son cruciales, especialmente cuando enviamos información sensible o valiosa. TLS utiliza una variedad de técnicas criptográficas para abordar cada uno de estos tres problemas. Juntos, permiten que el protocolo autenticar a la otra parte en una conexión, verificar la integridad de los datos y proporcionar protección cifrada.

TLS cumple estos requisitos utilizando una serie de procesos diferentes. Comienza con lo que se conoce como Apretón de manos TLS, que es donde se lleva a cabo la autenticación y se establecen las claves.

TLS vs. SSL

Al leer sobre TLS, a menudo verá mención de SSL o incluso como TLS / SSL. Secure Sockets Layer (SSL) es la versión anterior de TLS, pero muchos en la industria todavía se refieren a TLS bajo el antiguo nombre.

La historia de TLS

Todo comenzó con la necesidad de asegurar la capa de transporte. Como mencionamos anteriormente, el precursor de TLS fue SSL. Las primeras versiones de SSL fueron desarrolladas en los años noventa por Netscape, una compañía que construyó uno de los primeros navegadores web.

SSL 1.0 nunca se lanzó porque contenía vulnerabilidades graves. La versión 2.0 salió con Netscape Navigator 1.1 en 1995, sin embargo, todavía contenía una serie de fallas graves. SSL 3.0 fue

una versión fuertemente rediseñada y salió en 1996, con muchos de los problemas de seguridad resueltos.

En 1996, El IETF lanzó un borrador de SSL 3.0 en RFC 6101. El IETF formó un grupo de trabajo para estandarizar SSL, publicando los resultados en 1999 como TLS 1.0. Fue documentado en RFC 2246 y la estandarización incluyó algunos cambios al protocolo original, así como el cambio de nombre. Estas modificaciones se produjeron como resultado de negociaciones entre Netscape, Microsoft y el grupo de trabajo IETF.

En 2006, el IETF lanzó RFC 4346, que documenta TLS 1.1. Esta versión contenía nuevas disposiciones de seguridad y una serie de otras actualizaciones. La versión 1.2 se lanzó solo dos años más tarde en 2008. Incluyó soporte para cifrados de cifrado autenticado, una serie de cambios en la forma en que se utilizaron las funciones hash y muchas otras mejoras.

La próxima versión no llegó hasta 2018, cuando se definió TLS 1.3. Presenta una gran cantidad de cambios, incluido el secreto forzado, la eliminación de la compatibilidad con algoritmos más débiles y mucho más.

TLS: los detalles técnicos

TLS consta de muchos elementos diferentes. La parte fundamental es el protocolo de registro, el protocolo subyacente responsable de la estructura general de todo lo demás.

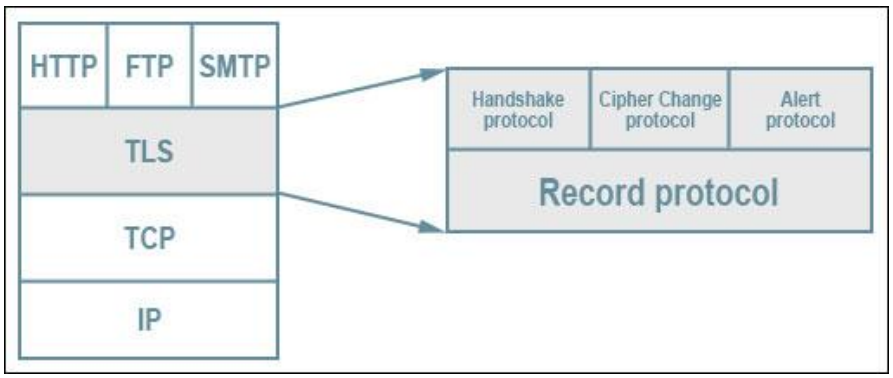

Figura 23: Diagrama que muestra la pila TLS. Pila de protocolos TLS de Jeffreytedjosukmono. Licenciado bajo CC0.

El protocolo de registro contiene cinco subprotocolos separados, cada uno de los cuales está formateado como registros:

HANDSHAKE o Apretón de manos - Este protocolo se utiliza para configurar los parámetros para una conexión segura.

Solicitud - El protocolo de aplicación comienza después del proceso de protocolo de enlace, y es donde los datos se transmiten de forma segura entre las dos partes..

Alerta - El protocolo de alerta es utilizado por cualquiera de las partes en una conexión para notificar a la otra si hay algún error, problemas de estabilidad o un posible compromiso..

Cambiar especificaciones de cifrado - Este protocolo lo utiliza el cliente o el servidor para modificar los parámetros de cifrado. Es bastante sencillo y no será cubierto en profundidad

HEARTBEAT o Latido del corazón - Esta es una extensión TLS que le permite a un lado de la conexión saber si su par aún está vivo y evita que los firewalls cierren conexiones inactivas. No es una parte central de TLS.

Cada uno de estos subprotocols se utiliza en diferentes etapas para comunicar información diferente. Los más importantes para entender son el protocolo de enlace y los protocolos de aplicación, ya que son responsables de establecer la conexión y luego transmitir los datos de forma segura.

El protocolo de enlace TLS

Aquí es donde se establece la conexión de manera segura.

Hay tres tipos básicos de protocolo de enlace TLS: el apretón de manos básico TLS, el protocolo de enlace TLS autenticado por el cliente y el apretón de manos abreviado.

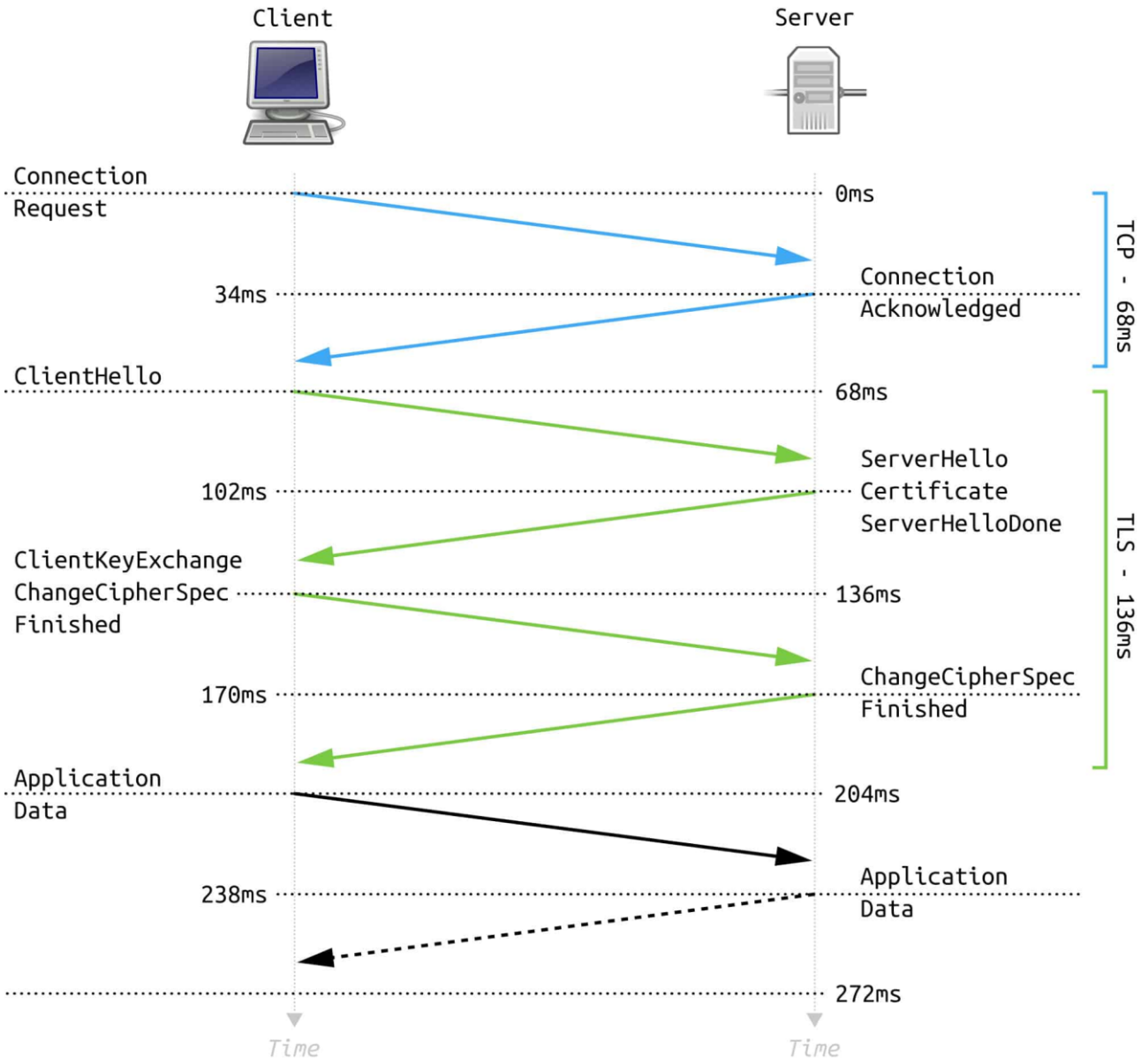

Figura 24: Diagrama que muestra el proceso de apretón de manos TLS. Apretón de manos completo TLS 1.2 por FleshGrinder. Licenciado bajo CC0.

El apretón de manos básico de TLS

En este tipo de protocolo de enlace, solo el servidor está autenticado y no el cliente. Comienza con la fase de negociación, donde un cliente envía un mensaje ClientHello. Contiene la versión más alta de TLS que admite el cliente, posibles conjuntos de cifrado, una indicación de si admite compresión, un número aleatorio y alguna otra información

El mensaje de saludo del cliente se encuentra con un mensaje ServerHello. Esta respuesta contiene el ID de sesión, la versión del protocolo, el conjunto de cifrado y la compresión (si se está utilizando) que el servidor seleccionó de la lista del cliente. También incluye un número aleatorio diferente..

Depende del conjunto de cifrado que se haya seleccionado, pero el servidor generalmente lo seguirá enviando un Certificado. Esto valida su identidad y contiene su clave pública.

Si se utilizan intercambios de claves efímeros de Diffie-Hellman o anónimos de Diffie-Hellman, esto es seguido por un Server Key Exchange mensaje. Otros métodos de intercambio de claves omiten esta parte. Cuando el servidor ha terminado con su lado de la negociación, envía un mensaje ServerHelloDone.

Ahora es el turno del cliente nuevamente. Dependiendo del conjunto de cifrado elegido, enviará un Intercambio de clave de cliente mensaje. Esto puede contener una clave pública o un secreto premaster, que se cifra con la clave pública del servidor.

Ambas partes usan los números aleatorios y el secreto premaster para crear un secreto maestro. Las claves se derivan del secreto maestro, que luego se utilizan para autenticar y cifrar las comunicaciones.

El cliente luego envía un mensaje ChangeCipherSpec. Esto le dice al servidor que los siguientes mensajes ahora se autenticarán y cifrarán.

El cliente luego sigue esto con un mensaje Finished, que está encriptado y también contiene un Código de autenticación de mensaje (MAC) para la autenticación. El servidor descifra este mensaje y verifica el MAC. Si alguno de estos procesos falla, entonces la conexión debe ser rechazada.

Ahora es el turno del servidor para enviar un mensaje ChangeCipherSpec, así como un mensaje Finished con el mismo contenido que el anterior. El cliente también intenta descifrar y verificar el contenido. Si todo esto se completa con éxito, el apretón de manos habrá finalizado. En este punto, se establece el protocolo de aplicación. Los datos se pueden intercambiar de forma segura de la misma manera que Terminado mensaje desde arriba, con autenticación y cifrado opcional.

Apretón de manos TLS autenticado por el cliente

Este protocolo de enlace es muy similar al protocolo de enlace básico de TLS, pero el cliente también está autenticado. La principal diferencia es que después de que el servidor envía su Certificado, también envía un Solicitud de certificado, solicitando el certificado del cliente. Una vez que el servidor termina, el cliente envía su certificado en un Certificado.

El cliente luego envía su mensaje ClientKeyExchange, al igual que en el apretón de manos básico de TLS. Esto es seguido por el mensaje Certificado de verificación, que incluye la firma digital del cliente. Como se calcula a partir de la clave privada del cliente, el servidor puede verificar la firma utilizando la clave pública que se envió como parte del certificado digital del cliente. El resto de Apretón de manos TLS autenticado por el cliente sigue la misma línea que el apretón de manos básico de TLS.

Apretón de manos TLS abreviado

Una vez que ya ha tenido lugar un apretón de manos, TLS permite que gran parte del proceso se corte mediante el uso de un apretón de manos abreviado. Estos apretones de manos utilizan la ID de sesión para vincular la nueva conexión a los parámetros anteriores.

Un apretón de manos abreviado permite a ambas partes reanudar la conexión segura con la misma configuración que se negoció anteriormente. Debido a que parte de la criptografía que normalmente está involucrada en el inicio de un apretón de manos puede ser bastante pesada en recursos computacionales, esto ahorra tiempo y facilita la conexión.

El proceso comienza con el mensaje ClientHello. Es muy parecido al mensaje anterior de saludo del cliente, pero también contiene el ID de sesión de la conexión anterior. Si el servidor conoce la ID de sesión, la incluye en su mensaje ServerHello. Si no reconoce la ID de la sesión, devolverá un número diferente y, en su lugar, tendrá que realizarse un apretón de manos TLS completo.

Si el servidor reconoce la ID de sesión, entonces el Certificado y Intercambio de llaves se pueden omitir . Los pasos ChangeCipherSpec especificaciones de cifrado y Finished se envían de la misma manera que el apretón de manos básico de TLS que se muestra arriba. Una vez que el cliente ha descifrado el mensaje y verificado el MAC, los datos se pueden enviar a través de la conexión segura TLS.

También hay una extensión TLS que permite reanudar las conexiones con tickets de sesión en lugar de identificadores de sesión. El servidor cifra los datos sobre la sesión y los envía al cliente. Cuando el cliente desea reanudar esta conexión, envía el ticket de sesión al servidor, que lo descifra para revelar los parámetros.

Los tickets de sesión no se usan con tanta frecuencia porque requieren el soporte de la extensión para gestionar tickets. A pesar de esto, pueden ser ventajosos en ciertas situaciones, porque el servidor no tiene que almacenar nada.

Análisis del handshake TLS

Los tres pasos más importantes del apretón de manos incluyen:

1. se seleccionan los parámetros,
2. se lleva a cabo la autenticación y
3. se establecen las claves

Vamos a cubrirlos con un poco más de detalle para que pueda entender lo que realmente está sucediendo..

Los parámetros

Al comienzo del apretón de manos, el cliente y el servidor negocian los parámetros de la conexión de común acuerdo. El primero de ellos es qué versión de TLS se utilizará. Esta es la versión más alta que ambas partes admiten, que tiende a ser la más segura.

Las partes también deciden qué algoritmo de intercambio de claves utilizarán para establecer la clave maestra. La función hash, el algoritmo de cifrado y el método de compresión también se acuerdan en esta etapa.

Autenticación: certificados digitales

La autenticación es una parte clave para asegurar un canal de comunicación, porque les permite a ambas partes saber que en realidad están hablando con quienes creen que son y no con un impostor. En TLS y muchos otros mecanismos de seguridad, esto se logra con los certificados digitales.

Establecer un secreto maestro

Como vimos anteriormente cuando discutimos el proceso de handshake básico de TLS, después de que una parte (o ambas partes) demuestre su identidad con su certificado público, el siguiente paso es establecer el secreto maestro, también conocido como secreto compartido. El secreto maestro es la base para derivar las claves utilizadas para cifrar y verificar la integridad de los datos transmitidos entre las dos partes.

El protocolo de enlace TLS puede usar varios mecanismos diferentes para compartir este secreto de forma segura. Estos incluyen RSA, varios tipos diferentes de intercambio de claves Diffie-Hellman, PSK, Kerberos y otros. Cada uno tiene sus propias ventajas y desventajas, como proporcionar confidencialidad directa, pero estas diferencias están fuera del alcance de esta unidad.

El proceso exacto dependerá del método de intercambio de claves que se haya elegido, pero sigue los pasos generales mencionados en la sección El apretón de manos básico de TLS

El secreto premaster se deriva de acuerdo con el método de intercambio de claves que se haya seleccionado previamente. El cliente cifra el secreto premaster con la clave pública del servidor para enviarlo de forma segura a través de la conexión.

El cliente y el servidor usan el secreto premaster y los números aleatorios que enviaron al comienzo de la comunicación para obtener el secreto maestro. Una vez que se ha calculado la clave maestra, se utiliza para obtener cuatro o seis claves separadas. Estos son los:

- Clave MAC cliente - El servidor utiliza esta clave para verificar la integridad de los datos enviados por el cliente.
- Clave MAC servidor - El cliente utiliza esta clave MAC de escritura del servidor para verificar la integridad de los datos enviados por el servidor.
- Clave de cifrado de envíos al cliente - El servidor usa esta clave para encriptar los datos enviados por el cliente.
- Clave de cifrado de de envíos al servidor - El cliente usa esta clave para encriptar los datos enviados por el servidor.
  Si el cifrado seleccionado corresponde a AEDD entonces se generan dos claves más

El establecimiento de la clave maestra es una parte importante del protocolo de enlace TLS, ya que permite que ambos lados de la conexión deriven claves de forma segura que se puedan usar tanto para la autenticación como para el cifrado. Se utilizan claves separadas para ambos procesos como medida de precaución.

Una vez que se han derivado las claves de autenticación y cifrado, se utilizan para proteger ambas Terminado mensajes, así como registros enviados a través del protocolo de aplicación.

El protocolo de aplicación

Una vez que el protocolo de enlace TLS ha establecido una conexión segura, el protocolo de aplicación se utiliza para proteger los datos transmitidos. Se puede configurar para usar una amplia gama de algoritmos para adaptarse a diferentes escenarios.

Algoritmos de autenticación

La integridad de los mensajes puede verificarse con muchos algoritmos diferentes. Éstos incluyen:

HMAC-MD5

HMAC-SHA1

HMAC-SHA2

AEAD

Algoritmos de cifrado

TLS utiliza cifrado de clave simétrica para proporcionar confidencialidad a los datos que transmite. A diferencia del cifrado de clave pública, solo se usa una clave en los procesos de cifrado y descifrado. Una vez que los datos se han cifrado con un algoritmo, aparecerá como una mezcla de texto cifrado. Mientras se utilice un algoritmo apropiado, los atacantes no podrán acceder a los datos reales, incluso si los interceptan..

TLS puede usar muchos algoritmos diferentes, como Camellia o ARIA, aunque el más popular es AES.

Compresión

La compresión es el proceso de codificación de datos para que ocupe menos espacio. TLS admite la compresión si ambos lados de la conexión deciden usarla. A pesar de esta capacidad, generalmente se recomienda evitar el uso de TLS para comprimir datos, especialmente desde el ataque CRIME (cuyo análisis excede los objetivos de este apunte) se descubrió que podía aprovechar los datos comprimidos para el secuestro de sesión.

Relleno

El relleno agrega datos adicionales a un mensaje antes de encriptarlo. Es un proceso criptográfico común que se utiliza para ayudar a evitar que las sugerencias en la estructura de los datos cifrados den su verdadero significado. TLS generalmente aplica el relleno PKCS # 7 a los registros antes de que se cifren.

Protocolo de alerta

Si la conexión o la seguridad se vuelven inestables, comprometidas o se ha producido un error grave, El protocolo de alerta permite al remitente notificar a la otra parte. Estos mensajes tienen dos tipos, ya sea de advertencia o fatales. Un mensaje de advertencia indica que la sesión es inestable y permite al destinatario determinar si la sesión debe continuar o no.

Un mensaje fatal le dice al destinatario que la conexión se ha visto comprometida o se ha producido un error grave. El remitente debe cerrar la conexión después de enviar el mensaje. El protocolo de alerta también contiene información sobre lo que está causando el problema de conexión particular. Esto puede incluir cosas como la falla de descifrado, una autoridad de certificación desconocida, un parámetro ilegal y mucho más.

TLS & el modelo OSI

El modelo OSI es una forma de conceptualizar y estandarizar cómo vemos nuestros diferentes sistemas y protocolos de comunicación. Es importante tener en cuenta que es solo un modelo, y algunos de nuestros protocolos no se ajustan a él..

Sin embargo, el OSI tiene siete capas separadas que muestran los niveles en los que operan los protocolos. TLS se encuentra sobre otro protocolo de transporte como TCP, sin embargo se usa más prominentemente como una capa de transporte, ya que las aplicaciones usan TLS en lugar de trasporte..

Uso de TLS

Puede proteger protocolos como HTTP, SMTP, FTP, XMPP y NNTP, tan bien como otros. La aplicación más común es el Protocolo seguro de transferencia de hipertexto (HTTPS), que protege la conexión entre un navegador web y un sitio web. Puede saber cuándo se utiliza HTTPS para proteger su conexión en línea, porque aparecerá un pequeño ícono de candado verde a la izquierda de la URL en la parte superior de su navegador.

TLS también se puede usar para construir VPN, como en OpenConnect y OpenVPN. Utiliza sus capacidades de encriptación y autenticación para formar un túnel que puede conectar hosts y redes entre sí. Las tecnologías VPN basadas en TLS como OpenVPN son ventajosas sobre alternativas como IPsec, porque no se sabe que OpenVPN tenga problemas de seguridad graves y pueden ser más fáciles de configurar.

Otro de sus usos es correo electrónico seguro a través de STARTTLS. Cuando se implementa TLS, evita que los atacantes puedan acceder a los mensajes mientras viajan entre los servidores de correo.

Problemas de seguridad de TLS

Como la mayoría de los protocolos, TLS ha tenido una serie de vulnerabilidades pasadas y ataques teóricos contra sus diversas implementaciones. A pesar de esto, las últimas versiones se consideran seguras para fines prácticos.

Las versiones anteriores, como SSL 2.0 y 3.0 (y TLS 1.0, que es esencialmente lo mismo que SSL 3.0) presentan numerosas fallas de seguridad, pero como son protocolos antiguos y obsoletos, no entraremos en detalles. Debería usar TLS 1.2 y 1.3 para proteger sus conexiones.

Las versiones más recientes de TLS tienen numerosas actualizaciones que lo hacen menos vulnerable que SSL. A pesar de esto, el protocolo aún ha tenido los siguientes problemas de seguridad, como por ejemplo HEARTBLEED

Heartbleed fue una falla de seguridad que se introdujo accidentalmente en la biblioteca de criptografía OpenSSL en 2012, pero no se publicitó hasta 2014. Debido a que esta es una implementación tan utilizada de TLS, causó un daño global significativo.

Uno de los desarrolladores de la extensión de latido TLS agregó una vulnerabilidad de sobre-lectura de búfer, que permite exponer algunos datos adicionales. El error no se detectó cuando se revisó el código, lo que condujo a una serie de ataques significativos.

Dado que la biblioteca OpenSSL se implementa tan ampliamente, el costo internacional de mitigar el problema terminó siendo bastante costoso. Los administradores del servidor tuvieron que instalar el nuevo parche y regenerar certificados y pares de claves que pueden haber estado en peligro durante el período de dos años que existió la vulnerabilidad.

11. Seguridad en la comunicación

EL PAPEL DE LAS REDES PRIVADAS VIRTUALES

Las organizaciones necesitan conectar sus centros de producción, oficinas centrales y puntos de venta para intercambiar datos en tiempo real a fin de llevar adelante sus actividades específicas. Para ello, se requieren enlaces dedicados que proporcionen un medio de comunicación confiable y seguro entre las distintas partes. No obstante, estas líneas dedicadas de una cierta capacidad tienen un costo muy elevado, por lo que sólo están al alcance de las grandes organizaciones.

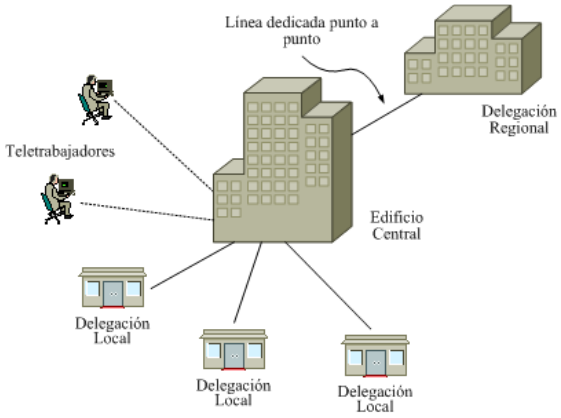

Figura 23: Red privada de una organización

Además, hoy en día muchos empleados necesitan acceder de forma remota a los recursos informáticos de la organización: teletrabajadores que realizan buena parte del trabajo desde sus hogares, comerciales que acceden a la información actualizada desde sus dispositivos portátiles, directivos que se encuentran de viaje y necesitan seguir conectados a la oficina central de la organización desde un hotel o una oficina remota, etc. Para todas estas situaciones resulta inviable establecer un enlace dedicado punto a punto.

Una **Red Privada Virtual** -VPN- (del idioma inglés *Virtual Private Network*) es un sistema de comunicaciones consistente en una red de datos restringida a un grupo cerrado de usuarios, que se construye empleando en parte o totalmente los recursos de una red de acceso público, es decir, es una extensión de la red privada de una organización usando una red de carácter público.

Una VPN constituye una alternativa económica y flexible para la conexión de empleados móviles, oficinas y delegaciones remotas a la red interna de la organización.

Al utilizar una VPN, las organizaciones pueden desentenderse de la complejidad y costos asociados a la conectividad telefónica y las líneas dedicadas punto a punto. Los usuarios de la organización simplemente se conectan al nodo geográficamente más cercano del operador de comunicaciones que ofrece se red pública para construir la VPN. Es este operador el que se encarga de la gestión de bancos de módems y servidores de comunicaciones, realizando el grueso de la inversión en tecnologías de acceso. Además, también cabe destacar la posibilidad de utilizar Internet para establecer la VPN de la organización, si bien en este caso no puede garantizarse la calidad de servicio y se incrementan los posibles problemas asociados con la seguridad de la conexión.

Para mejorar la seguridad del protocolo IP y facilitar la construcción de VPNs sobre Internet, la IETF ha desarrollado una nueva versión de IP (dentro del Proyecto IPv6), denominada *Internet Protocolo Security* -IPSec- en la RFC 4301. IPSec es una ampliación del protocolo IP que puede funcionar de modo transparente en las redes existentes, y que además permite establecer conexiones seguras con VPNs, mediante la creación de túneles seguros y garantizando la automatización de los dispositivos.

¿Por qué surge IPSec? La IETF ha sabido por años que hay una escasez de seguridad en Internet. Agregarla no era fácil pues surgió una controversia acerca de dónde colocarla. La mayoría de los expertos en seguridad creían que, para ofrecer una verdadera seguridad, el sistema de cifrado y las verificaciones de integridad tenían que llevarse a cabo de extremo a extremo (en la capa de aplicación). Esto es, el proceso de origen encripta o protege la integridad de los datos y los envía al proceso de destino, en donde se desencriptan o verifican. Por lo tanto, es posible detectar cualquier alteración que se realice entre estos dos procesos, o en cualquier sistema operativo. El problema con este enfoque es que requiere cambiar todas las aplicaciones para que estén conscientes de la seguridad. Desde esta perspectiva, el siguiente enfoque más conveniente es colocar el encriptado en la capa de transporte o en una nueva capa entre la capa de aplicación y la de transporte, con lo que se conserva el enfoque de extremo a extremo, pero no hay que cambiar las aplicaciones.

La perspectiva opuesta es que los usuarios no entiendan la seguridad y no sean capaces de utilizarla de manera correcta, así como que nadie desee modificar los programas existentes de ninguna forma, por lo que la capa de red debe autentificar o encriptar paquetes sin que los usuarios estén involucrados. Después de años de batallas encarnizadas, esta perspectiva ganó suficiente soporte como para definir un estándar de seguridad de capa de red. En parte, el argumento era que al tener el encriptado en la capa de red no se evitaba que los usuarios conscientes de la seguridad hicieran lo correcto, además de que en cierto punto ayuda a los usuarios no conscientes de la seguridad.

El resultado de esta guerra fue un diseño llamado IPsec (Seguridad IP, del inglés *IP security),* el cual se describe en los RFC 2401, 2402 y 2406, entre otros. No todos los usuarios desean encriptado (ya que exige muchos recursos computacionales); por esta razón, en lugar de hacerlo opcional, se decidió requerir siempre, pero permitir el uso de un algoritmo nulo. Este algoritmo nulo se describe y alaba por su simplicidad, facilidad de implementación y gran velocidad en el RFC 2410.

Antes de entrar en los detalles específicos de IPsec, demos un paso atrás y consideremos que es lo que implica proporcionar confidencialidad en la capa de red. Con la confidencialidad en la capa de red entre una pareja de entidades de red (por ejemplo, entre dos routers, entre dos hosts o entre un router y un host) la entidad emisora cifra las cargas útiles de todos los datagramas que envíe hacia la entidad receptora. La carga útil cifrada podría ser un segmento TCP, un segmento UDP, un mensaje ICMP, etc. Si dispusiéramos de tal servicio de la capa de red, todos los datos enviados de una entidad a

la otra (incluyendo los mensajes de correo electrónico, las páginas web, los mensajes de acuerdo TCP y los mensajes de administración, como ICMP y SNMP) estarían ocultos a ojos de posibles terceros que pudieran estar husmeando los mensajes que circulan por la red. Por esta razón, decimos que la seguridad de la capa de red proporciona un servicio básico de "ocultación".

Además de la confidencialidad, un protocolo de seguridad de la capa de red podría potencialmente proporcionar otros servicios de seguridad. Por ejemplo, podría ofrecer mecanismos de autenticación del origen de modo que la entidad receptora pueda verificar cuál es el origen del datagrama seguro. Un protocolo de seguridad de la capa de red podría proporcionar un servicio de integridad de los datos de modo que la entidad receptora pueda comprobar si se ha producido alguna alteración del datagrama mientras este se encontraba en tránsito. Un servicio de seguridad de la capa de red también podría proporcionar mecanismos para prevenir ataques por reproducción, lo que significa que Bob podría detectar cualquier datagrama duplicado que un atacante pudiera insertar. Como pronto veremos, IPsec de hecho proporciona mecanismos para todos estos servicios de seguridad, es decir, para la confidencialidad, la autenticación de origen, la integridad de los datos y la prevención de los ataques por reproducción.

IPsec y redes privadas virtuales (VPN)

Normalmente, una institución que abarque múltiples regiones geográficas desea disponer de su propia red IP, de modo que sus hosts y servidores puedan intercambiarse datos de forma segura y confidencial. Para conseguir este objetivo, esta institución podría implantar realmente una red física independiente (incluyendo routers, enlaces y una infraestructura DNS) que esté completamente separada de la red Internet pública. Dicha red separada, dedicada a una institución concreta, se denomina red privada. No es sorprendente que tales redes privadas puedan llegar a ser muy costosas, ya que la institución necesita comprar, instalar y mantener su propia infraestructura física de red.

Como ya hemos dicho antes, en lugar de implantar y mantener una red privada, muchas instituciones crean actualmente redes VPN sobre la red Internet pública existente. Con una VPN el tráfico entre sucursales se envía a través de la red Internet pública, en lugar de enviarse a través de una red físicamente independiente. Pero para proporcionar confidencialidad, el tráfico entre sucursales se cifra antes de entrar en la Internet pública. En la Figura 24 se muestra un ejemplo simple de red VPN. Aquí, la institución está compuesta por una oficina principal, una sucursal y una serie de vendedores itinerantes que suelen acceder a Internet desde la habitación de su hotel. (En la figura solo se muestra uno de esos vendedores.) En esta VPN, cuando dos hosts situados en la oficina principal se intercambian datagramas IP o cuando dos hosts de la sucursal quieren comunicarse utilizan el protocolo simple y tradicional IPv4 (es decir, sin servicios IPsec). Sin embargo, cuando dos hosts de la institución se comunican a través de una ruta que atraviesa la red Internet pública, el tráfico se cifra antes de entrar en Internet.

Para entender cómo funciona una red VPN, veamos un ejemplo simple en el contexto de la Figura 24. Cuando un host de la oficina principal envía un datagrama IP a un vendedor que se encuentra en un hotel, el router de pasarela de la oficina principal convierte el datagrama IPv4 simple en un datagrama IPsec y luego reenvía dicho datagrama IPsec hacia Internet. Los routers de la red Internet pública procesan el datagrama como si se tratara de un datagrama IPv4 normal; para ellos el datagrama es, de hecho, como cualquier otro. Pero como se muestra en la Figura 24, la carga útil del datagrama IPsec incluye una cabecera IPsec, que es utilizada para el procesamiento IPsec; además, la carga útil del datagrama IPsec está cifrada. Cuando el datagrama IPsec llega al portátil del vendedor, el

sistema operativo del equipo descifra la carga útil y proporciona algunos otros servicios de seguridad, como la verificación de la integridad de los datos, y pasa la carga útil descifrada hacia el protocolo de la capa superior (por ejemplo, hacia TCP o UDP).

Esto es solo una pequeña panorámica de como una institución podría utilizar IPsec para crear una red VPN. Para que los arboles no nos oculten el bosque, hemos dejado conscientemente de lado muchos detalles importantes. Realicemos ahora un examen más detallado.

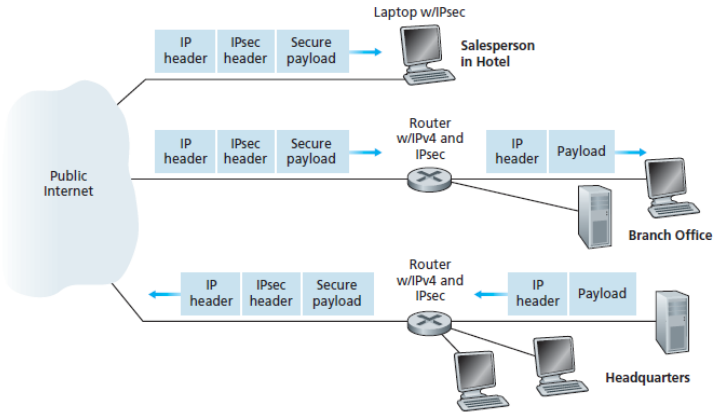

Figura 24: Red Privada Virtual (VPN)

IPSec

El diseño IPsec completo es un marco de trabajo para múltiples servicios, algoritmos y niveles de granularidad. La razón de los servicios múltiples es que no todas las personas quieren pagar el precio por tener todos los servicios todo el tiempo, por lo que están disponibles en todo momento. Los servicios principales son confidencialidad, integridad de datos y protección contra ataques de repetición (en donde un intruso repite una conversación). Todos estos servicios se basan en criptografía de clave simétrica debido a que es imprescindible un alto desempeño.

La razón de tener múltiples niveles de granularidad es para que sea posible proteger una sola conexión TCP, todo el tráfico entre un par de hosts o todo el tráfico entre un par de enrutadores seguros, entre otras posibilidades.

Un aspecto un poco sorprendente de IPsec es que, aun cuando se encuentra en la capa IP, es orientado a conexión. En la actualidad, eso no es tan sorprendente porque para tener seguridad, se debe establecer y utilizar una clave durante cierto período (en esencia, un tipo de conexión mediante un nombre distinto). Además, las conexiones amortizan los costos de configuración sobre muchos paquetes. Una “conexión” en el contexto de IPsec se conoce como SA (Asociación de Seguridad, del inglés *Security Association*). Una SA es una conexión simple entre dos puntos finales y tiene un identificador de seguridad asociado con ella. Si se necesita tráfico seguro en ambas direcciones, se requieren dos asociaciones de seguridad. Los identificadores de seguridad se transportan en paquetes que viajan en estas conexiones seguras, y se utilizan para buscar claves además de otra información relevante cuando llega un paquete seguro.

Técnicamente, IPSec tiene dos partes principales: la primera describe dos cabeceras nuevas que pueden agregarse a paquetes para transportar el identificador de seguridad, datos de control de integridad, entre otra información; la otra parte, tiene que ver con el establecimiento de las claves.

IPsec puede utilizarse en uno de dos modos:

Modo transporte, que permite establecer una comunicación segura extremo a extremo, ya que los propios dispositivos terminales utilizan el protocolo IPSec para encriptar los datos transmitidos. En este caso, no se cifra la cabecera de los paquetes IP, sino que la cabecera IPSec se inserta justo después de la cabecera IP. El campo *Protocol* de la cabecera IP se modifica para indicar que una cabecera IPSec sigue a la cabecera IP normal (antes de la cabecera de la capa de transporte, por ejemplo, TCP). El encabezado IPsec contiene información de seguridad, principalmente el identificador SA, un nuevo número de secuencia y tal vez una verificación de integridad del campo de carga

Modo túnel, estableciendo una comunicación segura entre ruteadores, en la que se lleva a cabo la encriptación de los paquetes IP (incluyendo su cabecera) y se les añade a continuación otra cabecera para facilitar su ruteo. Este modo de funcionamiento permite establecer túneles VPN sin que los dispositivos terminales tengan que emplear directamente el protocolo IPSec.

En el modo de túnel, todo el paquete IP, con encabezado y demás información, se encapsula en el cuerpo de un paquete IP nuevo con un encabezado IP totalmente nuevo. El modo de túnel es útil cuando termina en una ubicación que no sea el destino final. En algunos casos, el final del túnel es una máquina de puerta de enlace de seguridad; por ejemplo, el *firewall* de una empresa. Éste es el caso común para una VPN (Red Privada Virtual). En este modo, la puerta de enlace de seguridad encapsula y desencapsula paquetes conforme pasan a través de ella. Al terminar el túnel en esta máquina segura, las máquinas en la LAN de la empresa no tienen que estar al tanto de IPsec. Sólo la puerta de enlace de seguridad tiene que saber acerca de ello.

El modo de túnel también es útil cuando se agrega un conjunto de conexiones TCP y se maneja como un solo flujo cifrado, porque así se evita que un intruso vea quién está enviando cuántos paquetes a quién. Algunas veces el simple hecho de saber cuánto tráfico está pasando y hacia dónde se dirige es información valiosa. Por ejemplo, si durante una crisis militar, la cantidad de tráfico que fluye entre el Pentágono y la Casa Blanca se reduce de manera considerable, pero la cantidad de tráfico entre el Pentágono y alguna instalación militar oculta entre las Montañas Rocosas de Colorado se incrementa en la misma proporción, un intruso podría ser capaz de deducir alguna información útil a partir de estos datos. El estudio de los patrones de flujo de los paquetes, aunque estén encriptados, se conoce como análisis de tráfico. El modo de túnel proporciona una forma de frustrarlo hasta cierto punto. La desventaja del modo de túnel es que agrega un encabezado IP adicional, por lo que se incrementa el tamaño del paquete en forma considerable. En contraste, el modo de transporte no afecta tanto al tamaño del paquete.

Los protocolos AH y ESP

IPsec es un protocolo bastante complejo que está definido en más de una docena de documentos RFC. Dos documentos importantes son RFC 4301, que describe la arquitectura global de seguridad IP, y RFC 2411, que proporciona una panoramica de la serie de protocolos IPsec.

En la serie de protocolos IPsec hay dos protocolos principales: el protocolo de Cabecera de autenticación (AH, *Authentication Header)* y el protocolo de Carga útil de seguridad para encapsulación (ESP, *Encapsulation Security Payload).* Cuando una entidad IPsec de origen (normalmente un host o un router) envía datagramas seguros a una entidad de destino (también un host o un router) lo hace con el protocolo AH o el protocolo ESP. El protocolo AH proporciona

autenticación del origen e integridad de los datos, pero *no* proporciona confidencialidad. El protocolo ESP proporciona autenticación del origen, integridad de los datos *y* confidencialidad.

Dado que ESP puede hacer lo mismo que AH y más, y debido a que es más eficiente en el arranque, surge la pregunta: ¿por qué molestarse en tener a AH? La respuesta es en su mayor parte por cuestiones históricas. En un principio, AH se encargaba sólo de la integridad y ESP sólo de la confidencialidad. Más tarde se agregó la integridad a ESP, pero las personas que diseñaron AH no querían dejarlo morir después de todo el trabajo que habían realizado. Su único argumento válido es que AH verifica parte del encabezado IP, lo cual ESP no hace, pero es un argumento débil. Otro argumento débil es que un producto que soporta AH y no ESP podría tener menos problemas para obtener una licencia de exportación, porque no puede realizar la encriptación. Es probable que AH sea desplazado en el futuro. Puesto que la confidencialidad a menudo es crítica para las redes VPN y otras aplicaciones IPsec y con el fin de desmitificar IPsec y evitar buena parte de las complicaciones asociadas nos vamos por tanto a centrar en el protocolo ESP.

Asociaciones de seguridad

Antes de enviar datagramas IPsec desde la entidad de origen a la de destino, ambas entidades crean una conexión lógica en la capa de red. Esta conexión lógica se denomina asociación de seguridad (SA, *Security Association).* Una asociación de seguridad es una conexión lógica de tipo simplex; es decir, una conexión unidireccional desde el origen al destino. Si ambas entidades desean enviarse datagramas seguros entre sí, entonces será necesario establecer dos SA (es decir, dos conexiones lógicas), una en cada dirección. Por ejemplo, considere de nuevo la VPN institucional de la Figura 24. Esta institución consta de una oficina principal, una sucursal y un cierto número, por ejemplo, *n,* de vendedores itinerantes. Supongamos, como ejemplo, que existe tráfico IPsec bidireccional entre la oficina principal y la sucursal y entre la oficina principal y los vendedores. En esta VPN, ¿cuántas asociaciones de seguridad existirían? Para responder a esta cuestión, observe que hay dos SA entre el router de pasarela de la oficina principal y el router de pasarela de la sucursal (una en cada dirección); para la computadora portátil de cada vendedor también habrá dos SA entre el router de pasarela de la oficina principal y el portátil (de nuevo, una en cada dirección). Por tanto, en total, habrá (2 + 2*n)* asociaciones de seguridad. *Sin embargo, recuerde que no todo el tráfico enviado hacia Internet por los routers de pasarela o por las computadoras portátiles estará protegido mediante IPsec.* Por ejemplo, un host situado en la oficina principal podría querer acceder a un servidor web (como Amazon o Google) disponible en la red Internet pública. Por tanto, el router de pasarela (y los portátiles) enviara hacia Internet tanto datagramas IPv4 normales como datagramas dotados de seguridad IPsec.

Tratemos ahora de examinar las interioridades de una asociación de seguridad. Para que las explicaciones sean tangibles y concretas vamos a hacerlo en el contexto de una asociación de seguridad existente entre el router R1 y el router R2 de la Figura 25. (Podemos considerar que el router R1 es el router de pasarela de la oficina principal y que el router R2 es el router de pasarela de la sucursal en el contexto de la Figura 24.) El router R1 mantendrá una cierta información de estado acerca de esta SA, la cual incluirá:

• Un identificador de 32 bits para la SA, denominado índice de parámetro de seguridad (SPI, *Security Parameter Index).*

• La interfaz de origen de la SA (en este caso, 200.168.1.100) y la interfaz de destino de la SA (en este caso 193.68.2.23).

- El tipo de cifrado que se va a utilizar (por ejemplo, 3DES con CBC).
- La clave de cifrado.
- El tipo de comprobación de integridad (por ejemplo, HMAC con MD5).
- La clave de autenticación.

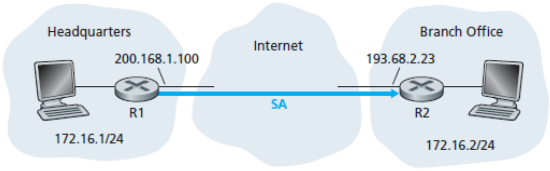

Figura 25: Asociación de seguridad (SA) de R1 a R2.

Cada vez que el router R1 necesite construir un datagrama IPsec para reenviarlo a través de esta SA, accederá a esta información de estado para determinar cómo debe autenticar y cifrar el datagrama. De forma similar, el router-R2 mantendrá la misma información de estado para esta SA y utilizará esta información para autenticar y descifrar todos los datagramas IPsec que lleguen desde dicha asociación de seguridad. Cada entidad IPsec (router o host) suele mantener información de estado para muchas asociaciones de seguridad. Por ejemplo, en el ejemplo de la red VPN de la Figura 24 con *n* vendedores, el router de pasarela de la oficina principal mantiene información de estado para (2 + 2*n)* asociaciones de seguridad. Cada entidad IPsec almacena la información de estado para todas sus asociaciones de seguridad en su Base de datos de asociaciones de seguridad (SAD, *Security Association Database),* que es una estructura de datos contenida en el kernel del sistema operativo de esa entidad.

El datagrama IPsec

Habiendo descrito las asociaciones de seguridad, podemos ahora describir la estructura real del datagrama IPsec. IPsec tiene dos formas distintas de paquete, una para el denominado modo túnel y otra para el denominado modo transporte. El modo túnel, al ser más apropiado para las redes VPN, está más ampliamente implantado que el modo transporte. Con el fin de desmitificar todavía más IPsec y evitar buena parte de los aspectos más complejos, nos vamos a centrar por tanto exclusivamente en el modo túnel.

El formato de paquete del datagrama IPsec se muestra en la Figura 26. Examinemos los campos IPsec en el contexto de la Figura 25. Suponga que el router R1 recibe un datagrama IPv4 normal procedente del host 172.16.1.17 (situado en la red de la oficina principal) que está destinado al host 172.16.2.48 (situado en la red de la sucursal). El router R1 utiliza la siguiente receta para convertir este "datagrama IPv4 original" en un datagrama IPsec:

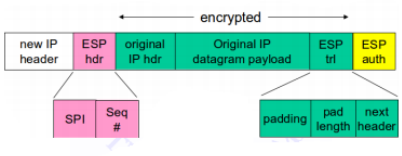

Figura 26: Formato del datagrama IPsec

• Añade al final del datagrama IPv4 original (¡que incluye los campos originales de cabecera!) un campo de "cola ESP".

• Cifra el resultado utilizando el algoritmo y la clave especificados por la asociación de seguridad.

• Añade al principio de este paquete cifrado un campo denominado "cabecera ESP", formando la "*carga útil sin autenticación*"

• Crea un valor MAC de autenticación al paquete cifrado + la cabecera ESP utilizando el algoritmo y la clave especificados en la SA.

• Añade el valor MAC al final formando así la *carga útil.*

• Por último, crea una nueva cabecera IP con todos los campos clásicos de la cabecera IPv4 (que suman normalmente 20 bytes de longitud) y añade dicha cabecera al principio de la carga útil.

Observe que el datagrama IPsec resultante es un datagrama IPv4 perfectamente normal, con los campos tradicionales de cabecera IPv4 seguidos de una carga útil. Pero en este caso la carga útil contiene una cabecera ESP, el datagrama IP original, una cola ESP y un campo de autenticación ESP (estando cifrados el datagrama original y la cola ESP). El datagrama IP original tiene el valor 172.16.1.17 como dirección IP de origen y el 172.16.2.48 como dirección IP de destino. Puesto que el datagrama IPsec incluye el datagrama IP original, estas direcciones se incluyen (y se cifran) como parte de la carga útil del paquete IPsec.

¿Pero qué sucede con las direcciones IP de origen y de destino contenidas en la nueva cabecera IP, es decir, en la cabecera situada más a la izquierda en el datagrama IPsec? Como cabría esperar, esos valores se configuran con las direcciones de las interfaces de router de origen y de destino situadas en los dos extremos de los túneles, es decir, con los valores 200.168.1.100 y 193.68.2.23. Asimismo, el número de protocolo en este nuevo campo de cabecera IPv4 no se configura con el valor correspondiente a TCP, UDP o SMTP, sino con el valor 50, que indica que se trata de un datagrama IPsec que está empleando el protocolo. Después de que R1 envíe el datagrama IPsec hacia la red Internet pública, pasará a través de muchos routers antes de alcanzar R2. Cada uno de estos routers procesará el datagrama como si fuera un datagrama normal; de hecho, todos esos routers no son conscientes de que el datagrama este transportando datos cifrados mediante IPsec. Para estos routers de ESP, la red Internet pública, puesto que la dirección IP de destino contenida en la cabecera externa es R2, el destino último del datagrama es R2.

Habiendo examinado este ejemplo de cómo se construye un datagrama IPsec, veamos ahora con más detalle los ingredientes de la carga útil. Como podemos ver en la Figura 26, la cola ESP está compuesta por tres campos: relleno, longitud de relleno y siguiente cabecera. Recuerde que los sistemas de cifrado de bloque requieren que el mensaje que hay que cifrar sea un múltiplo entero de la longitud de bloque. Por ello se emplea un relleno (compuesto por bytes que no tienen ningún significado) para que, al añadirlo al datagrama original (junto con los campos de longitud de relleno y de siguiente cabecera), el "mensaje" resultante tenga un número entero de bloques. El campo de longitud de relleno indica a la entidad receptora cuanto relleno se ha insertado (y, por tanto, cuanto relleno habrá que eliminar).

El campo de siguiente cabecera indica el tipo (por ejemplo, UDP) de los datos contenidos en el campo de datos de carga útil. Los datos de carga útil (normalmente, el datagrama IP original) y la cola ESP se concatenan y se cifran. Delante de esta unidad cifrada se encuentra la cabecera ESP, que se envía como texto en claro y que consta de dos campos: el SPI y el campo de número de secuencia. El SPI indica a la entidad receptora cuál es la SA a la que pertenece el datagrama; la entidad receptora puede entonces indexar su base de datos SAD con el índice SPI para determinar los algoritmos y claves apropiados de autenticación/descifrado. El campo de número de secuencia se utiliza para defenderse frente a los ataques por reproducción.

La entidad emisora también añade un código MAC de autenticación. Como hemos dicho anteriormente, la entidad emisora calcula un código MAC para toda la *carga útil sin autenticación*, cabecera ESP, el datagrama IP original y la cola ESP, estando el datagrama y la cola cifrados. Recuerde que para calcular un valor MAC, el emisor añade una clave secreta MAC a los campos antes detallados y luego calcula un valor hash de longitud fija para el resultado.

Cuando R2 recibe el datagrama IPsec, observa que la dirección IP de destino del datagrama es el propio R2, por lo que dicho router se encarga de procesar el datagrama. Puesto que el campo de protocolo (en la cabecera IP situada más a la izquierda) tiene el valor 50, R2 ve que debe aplicar el procesamiento ESP de IPsec al datagrama. En primer lugar, analizando la *carga útil sin autenticación*, R2 utiliza el SPI para determinar a qué asociación de seguridad (SA) pertenece el datagrama. En segundo lugar, calcula el valor MAC y verifica que es coherente con el valor contenido en el campo ESP MAC. Si lo es, el router sabrá que el datagrama procede del router R1 y que no ha sido manipulada. En tercer lugar, comprueba el campo de número de secuencia para verificar que el datagrama sea reciente y no un datagrama reproducido. En cuarto lugar, descifra la unidad cifrada utilizando la clave y el algoritmo de descifrado asociado con la SA. En quinto lugar, elimina el relleno y extrae el datagrama IP normal original. Y, finalmente, en sexto lugar, reenvía el datagrama original a la red de la sucursal para que el datagrama llegue a su verdadero destino.

Existe todavía otra sutileza importante que necesitamos explicar y que está centrada en la siguiente cuestión: cuando el router R1 recibe un datagrama (no dotado de seguridad) procedente de un host de la red de la oficina principal y dicho datagrama está destinado a alguna dirección IP de destino situada fuera de la oficina principal, ¿cómo sabe R1 si ese datagrama debe ser convertido en un datagrama IPsec? Y si tiene que ser procesado por IPsec, ¿cómo sabe R1 que SA (de las muchas asociaciones de seguridad existentes en su base de datos SAD) hay que utilizar para construir el datagrama IPsec? El problema se resuelve de la forma siguiente. Junto con una base de datos SAD, la entidad IPsec también mantiene otra estructura de datos denominada **Base de datos de políticas de seguridad (SPD, *Security Policy Database).*** La SPD indica qué tipos de datagramas (en función de la dirección IP de origen, la dirección IP de destino y el tipo de protocolo) hay que procesar mediante IPsec; y para aquellos que haya que procesar mediante IPsec, qué SA debe emplearse. En un cierto sentido, la información de una SPD indica “qué” hacer con los datagramas que lleguen, mientras que la información de la SAD indica “cómo” hay que hacerlo.

Resumen de los servicios IPsec

¿Qué servicios proporciona IPsec exactamente? Examinemos estos servicios desde la perspectiva de un atacante, como por ejemplo Trudy, que se ha interpuesto (*man-in-the-middle*) en la comunicación, situándose en algún lugar de la ruta entre los routers R1 y R2 de la Figura 25. Vamos a suponer a lo largo de estas explicaciones que Trudy no conoce las claves de cifrado y de autenticación

empleadas por la SA. ¿Qué cosas puede hacer Trudy y cuáles no? En primer lugar, Trudy no puede ver el datagrama original. De hecho, no solo están los datos del datagrama original ocultos a ojos de Trudy, sino que también lo están el número de protocolo, la dirección IP de origen y la dirección IP de destino. Para los datagramas enviados a través de la SA, Trudy solo sabe que el datagrama tiene su origen en algún host de la red 172.16.1.0/24 y que está destinado a algún host de la red 172.16.2.0/24. No sabe si está transportando datos TCP, UDP o ICMP; no sabe si está transportando HTTP, SMTP, o algún otro tipo de datos de aplicación. Esta confidencialidad, por tanto, va bastante más allá que en SSL. En segundo lugar, suponga que Trudy trata de alterar un datagrama en la SA modificando algunos de sus bits. Cuando este datagrama alterado llegue a R2 no pasara las comprobaciones de integridad (utilizando el valor MAC), desbaratando una vez más las intenciones de Trudy. En tercer lugar, suponga que Trudy intenta hacerse pasar por R1, creando un datagrama IPsec cuyo origen sea 200.168.1.100 y cuyo destino sea 193.68.2.23. El ataque de Trudy no tendrá ningún efecto, ya que este datagrama de nuevo no pasará la comprobación de integridad realizada en R2. Finalmente, puesto que IPsec incluye números de secuencia, Trudy no podrá desarrollar con éxito ningún ataque por reproducción. En resumen, y tal como dijimos al principio de esta sección, IPsec proporciona (entre cualquier pareja de dispositivos que procesen paquetes en la capa de red) mecanismos de confidencialidad, de autenticación del origen, de integridad de los datos y de prevención de los ataques por reproducción.

L2TP/IPsec: ¿qué es esto?

L2TP (Layer 2 Tunneling Protocol) es un protocolo utilizado para VPN que fue diseñado por un grupo de trabajo de IETF, como el heredero de PPTP, y fue creado para corregir las deficiencias de este protocolo y establecerse como un estándar. L2TP utiliza PPP para proporcionar acceso telefónico, que puede ser dirigido a través de un túnel por Internet hasta un punto determinado. L2TP incluye los mecanismos de autenticación de PPP, PAP y CHAP, además, de forma similar a PPTP, soporta la utilización de estos protocolos de autenticación, como RADIUS.

A pesar de que L2TP ofrece un acceso con soporte multiprotocolo y acceso a redes de área local remotas, no presenta unas características criptográficas especialmente robustas. Sólo se realiza la operación de autenticación entre los puntos finales del túnel, pero no para cada uno de los paquetes que viajan por él. Esto puede dar lugar a suplantaciones de identidad en algún punto interior al túnel. Sin comprobación de la integridad de cada paquete, sería posible realizar un ataque de denegación del servicio por medio de mensajes falsos de control que den por acabado el túnel L2TP o la conexión PPP subyacente.

L2TP no cifra de manera robusta el tráfico de datos de los usuarios, por tanto, da problemas cuando sea importante mantener la confidencialidad de los datos. A pesar de que la información contenida en los paquetes puede ser cifrada, este protocolo no dispone de mecanismos para generación automática de claves, o refresco automático de claves. Esto puede hacer que alguien que escuche en la red y descubra una única clave tenga acceso a todos los datos transmitidos.

Teniendo en cuenta todas estas debilidades de L2TP, la IETF tomó la decisión de utilizar los propios protocolos del protocolo IPsec, para proteger los datos que viajan por el túnel L2TP. Por este motivo, siempre se escriben de forma «L2TP/IPsec», por se hace uso de ambos protocolos simultáneamente, además, este protocolo conjunto es ampliamente utilizado. Se podría decir que L2TP es el protocolo a nivel de capa de “enlace”, y que no tiene seguridad, sin embargo, IPSec le proporciona la seguridad a nivel de capa de red para que la utilización de este protocolo sí sea segura.

Por este motivo, siempre vamos a encontrar la nomenclatura L2TP/IPSec de manera conjunta, porque se utilizan ambos protocolos para tener una conexión VPN segura.

12. Bibliografía

- FUNDAMENTOS DE SEGURIDAD EN REDES APLICACIONES Y ESTÁNDARES 2da Edición – William Stallings - Person Educación- 2004 – ISBN 84-205-4002-1

- REDES DE COMPUTADORAS. 5ta. Edición - Andrew S. Tanenbaum y David J. Wetherall - Person Education- 2012- ISBN: 978-607-32-0817-8

- COMUNICACIONES Y REDES DE COMPUTADORAS - 7ma edición - William Stallings - Pearson Eduación S.A – 2004- ISBN: 978-84-205-4110-5