Criptografía

Tema 2

Álvaro Rodríguez - IES Alonso de Madrigal (Ávila)
2º SMR

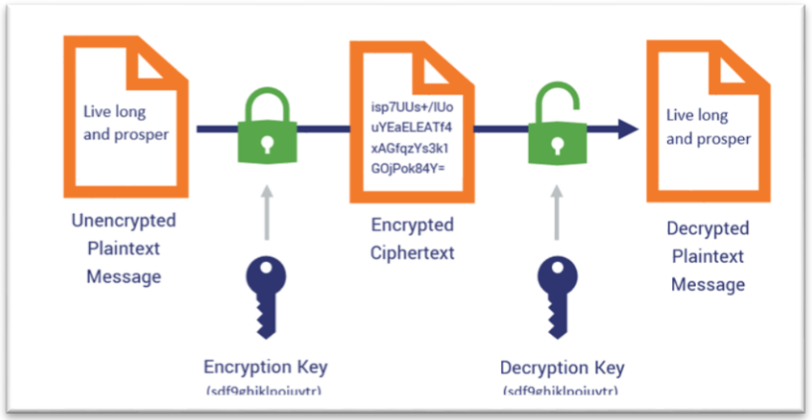
Criptografía

¿PARA QUÉ SIRVE LA CRIPTOGRAFÍA?

La palabra criptografía viene del griego **cripto** (que significa «ocultar») y **graphos** (que significa «escribir»). Se podría traducir por: cómo escribir mensajes ocultos

La criptografía consiste en tomar un dato en texto plano (legible) y aplicarle un algoritmo cuyo resultado es un dato encriptado (ilegible). Ese dato nuevo, al estar cifrado, podemos hacerlo llegar hasta el destinatario.

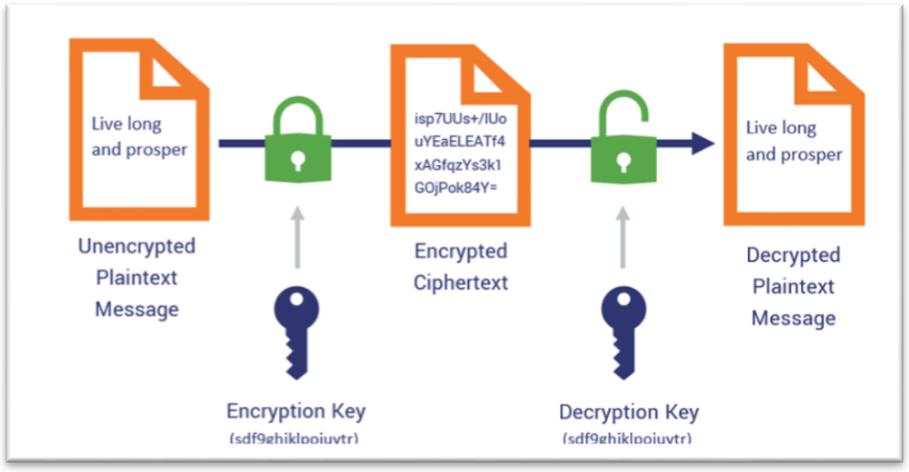
Una vez en el destino, se aplicará el algoritmo inverso para recuperar el dato original, de nuevo en texto plano



Criptografía

CÓMO FUNCIONA LA CRIPTOGRAFÍA

La privacidad la conseguimos gracias a la **clave** (key) del **algoritmo**: un conjunto de valores que, combinados con el documento original tal y como se indica en el algoritmo, generan un documento cifrado de tal forma que, solo con ese documento, es imposible deducir ni el documento original ni la clave utilizada. Por supuesto, debemos evitar que el enemigo pueda llegar a conocer nuestra clave.



Criptografía

TRABAJO DE CLASE

Comprueba que la conexión bajo HTTPS va cifrada utilizando el sniffer Wireshark

1. Configura tu adaptador de red para que esté en modo NAT o red NAT. ¿Qué diferencia tiene con el resto de las configuraciones?
2. Comprueba si tienes instalado Wireshark en tu MV. Instala el programa si no lo tuvieras.
3. Comprueba que adaptadores de red tienes en tu máquina y cuál es el que tiene conexión a Internet.
4. Captura los paquetes que pasen por el adaptador de red de tu MV que tiene conexión a Internet
5. Utiliza CURL para:
    - Hacer una petición a http://avila.es
    - Hacer una petición a https://avila.es
6. ¿Existe alguna diferencia con respecto a lo que recibes desde curl?
7. Localiza en cada caso, en Wireshark, la respuesta en HTML del servidor. ¿Qué diferencias encuentras?

Aspectos importantes de esta actividad: modos de configuración de red en MV, comprobación de programas instalados en Linux, instalación en Linux, protocolos que intervienen al navegar por la web (DNS, TCP, TSL…), Cliente-Servidor, CURL, Wireshark, Protocolos HTTP/HTTPS, códigos de respuesta en HTTP, protocolos de cifrado en HTTPS…

Criptografía

VULNERABILIDADES

La criptografía es un método que se utiliza para aportar seguridad, pero:

- Los propios métodos o algoritmos de cifrado pueden ser vulnerables.
- Aunque usemos un método vulnerable, siempre va a ser más seguro que no utilizar ningún tipo de cifrado

Ejemplos donde la criptografía es beneficiosa pero no suficente:

[El cifrado de extremo a extremo en Instagram y FB Messenger tendrá que esperar: su llegada se retrasa a 2023 (xataka.com)]

[Vulnerabilidades de criptografía en redes inalámbricas]

Criptografía

VULNERABILIDADES

Por ejemplo, nuestra seguridad está expuesta a los **ataques de fuerza bruta**: probar todas las combinaciones posibles de símbolos para lograr obtener el valor de una clave. Para evitarlo tomaremos estas medidas:

- Utilizar claves de gran **longitud**, de modo que el atacante necesite muchos recursos y mucho tiempo para cubrir todas las combinaciones.
- Utilizar todos los **tipos de caracteres o símbolos** posibles: una clave compuesta solo de números (diez valores posibles) es más fácil de adivinar que una con números y letras (36 valores posibles).
- No utilizar **palabras** fácilmente identificables: palabras de diccionario, nombres propios, etc.
- Cambiar regularmente la clave. De esta forma, si alguien quiere intentar cubrir todo el rango de claves, le limitamos el tiempo para hacerlo.
- Detectar repetidos intentos fallidos en un corto intervalo de tiempo. Por ejemplo, la tarjeta del móvil se bloquea si fallamos tres veces al introducir el PIN.

Las claves no son el único punto débil de la criptografía; pueden existir vulnerabilidades en el propio algoritmo o en su implementación en el software. Estas vulnerabilidades las estudia el criptoanálisis

Criptografía

TRABAJO DE CLASE

- ✓ Para cada una de las siguientes contraseñas, indica qué errores comete con respecto a la política de contraseñas recomendada:
    1. *Paco2000*
    2. *1qaz2wsx3ed*
    3. *zapato23*
    4. *1234567890*
    5. *l.*2s4*
    6. *holidays8.*
    7. *A#f98,mo0*
    8. *Adr1an*
    9. *Afpklhj*
    10. *1q2w3e4r*

- ✓ *Utiliza HaveiBeenPwned para saber si esa contraseña se ha filtrado en algún LEAK conocido de algún servicio web y cuántas veces aparece. ¿Hay alguna que te llame la atención por las apariciones que tiene?*

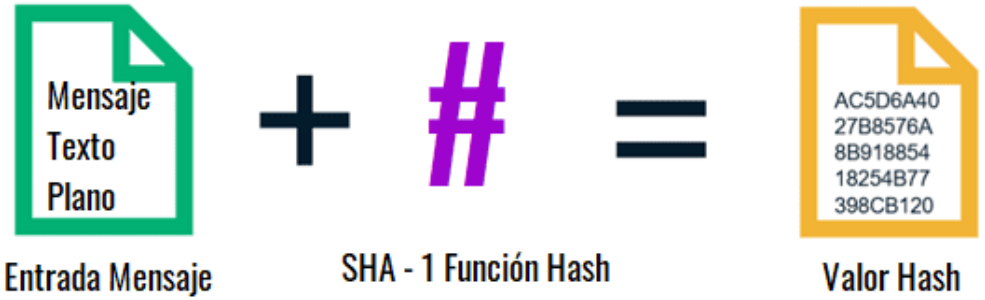
- ✓ *¿Crees que es buena idea utilizar una contraseña que tiene varias apariciones en LEAKS? ¿Por qué?*

- ✓ *Si tuvieras que escoger la mejor contraseña de las diez ¿cuál crees que sería? ¿Por qué?*

EJEMPLO DE CIFRADO: LOS HASHES

Un hash es un valor que se obtiene al aplicar un algoritmo de hashing sobre un valor de entrada (un número, un teto, un archivo…).

- Muy utilizado, por ejemplo, para no almacenar contraseñas en texto claro.
- Cuando el valor de entrada cambia, aunque sea mínimamente (1 bit), el hash cambia completamente. Por eso se utiliza para comprobar la integridad de datos
- A partir del hash no se puede obtener el valor original (al menos, no existe un algoritmo inverso al de hashing que lo haga)
- Existen muchos algoritmos de hashing:
  - Inseguros:MD5, SHA-1
  - Menos inseguros: SHA-256, SHA512, PDKF, bcrypt
- Sin embargo: vulnerable a fuerza bruta, rainbow tables, lista de hashes conocidos…



Criptografía

TRABAJO DE CLASE

1. Encuentra los hashes MD5 y SHA-256 de tu nombre utilizando:
    - md5sum y sha256sum en Linux
    - Get-FileHash en Poweshell
    - Alguna herramienta para obtener estos hashes de manera online

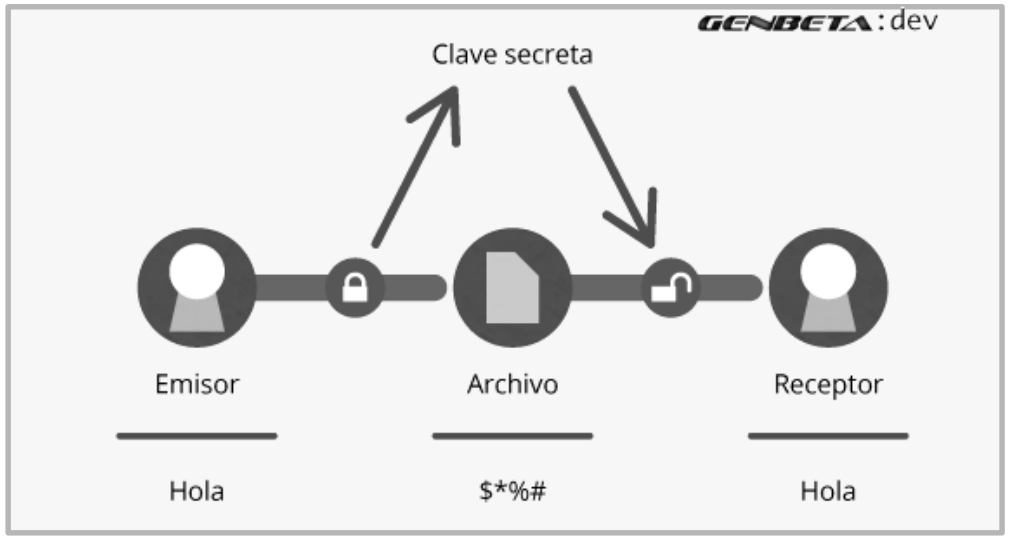
Compara los resultados. Deberías obtener los mismos Hashes.

2. Investiga sobre base64: ¿Es un algoritmo de hashing? ¿Por qué?
   Obtén tu nombre mediante base64.

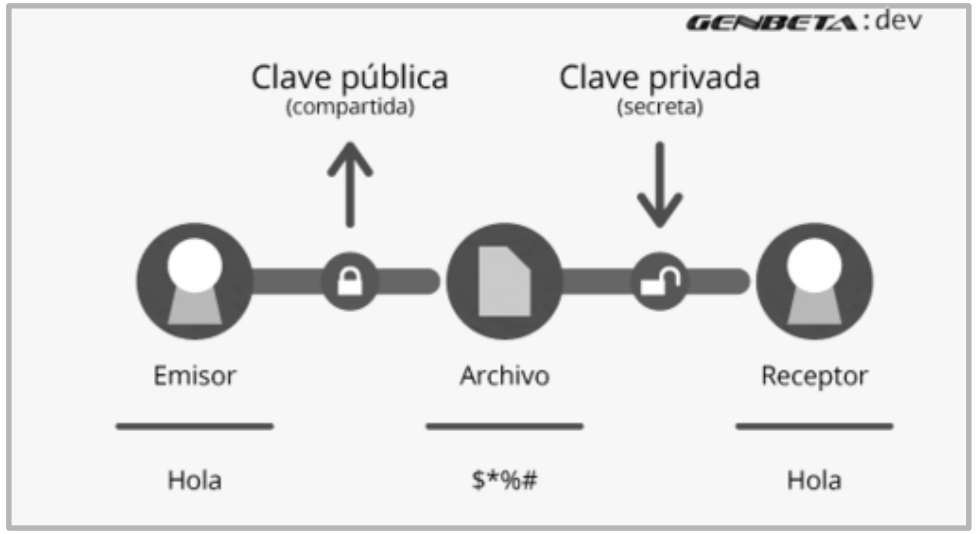
Criptografía

CIFRADO SIMÉTRICO Y ASIMÉTRICO

Existen dos tipos de cifrado distintos:



Simétrico



Asimétrico

https://www.youtube.com/watch?v=n04e6Zni4zA

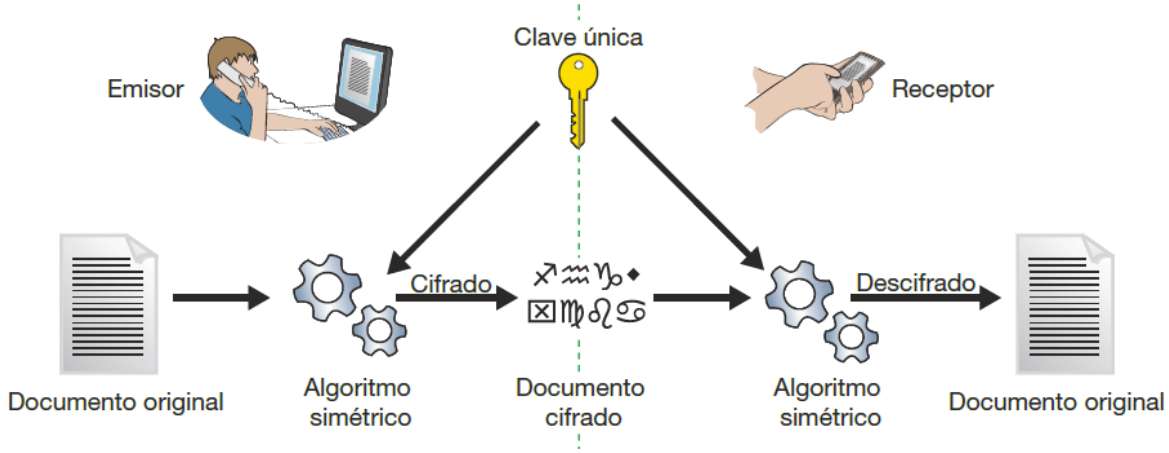
Criptografía

CIFRADO SIMÉTRICO

Los algoritmos de criptografía simétrica **utilizan la misma clave para los dos procesos: cifrar y descifrar.**

Son sencillos de utilizar y, en general, resultan **bastante eficientes (tardan poco tiempo en cifrar o descifrar)**
Los más utilizados actualmente son DES, 3DES, **AES**, Blowfish e IDEA

El funcionamiento es simple: el emisor quiere hacer llegar un documento al receptor. Toma ese documento y le aplica el algoritmo simétrico, usando la clave única, que también conoce el receptor. El resultado es un documento cifrado que ya podemos enviar tranquilamente. Cuando el receptor recibe este documento cifrado, le aplica el mismo algoritmo con la misma clave, pero ahora en función de descifrar. Si el documento cifrado no ha sido alterado en el camino y la clave es la misma, el resultado será el documento original



CIFRADO SIMÉTRICO

El cifrado simétrico tiene dos problemas principales, que serán los que venga a solucionar, precisamente, la criptografía asimétrica:

El problema principal la **circulación de las claves**: cómo conseguimos que el emisor y el receptor tengan la clave correcta. Necesitan un canal seguro para comunicársela, pero ¿qué método utilizamos?.

El segundo problema es la **gestión de las claves almacenadas**. Si en una empresa hay diez trabajadores y todos tienen conversaciones privadas con todos, cada uno necesita establecer nueve claves distintas y encontrar nueve canales seguros para actualizarlas cada vez (en total 81 claves y 81 canales).

Criptografía

TRABAJO DE CLASE

Crackeo de contraseña en un archivo .ZIP cifrado utilizando John The Ripper

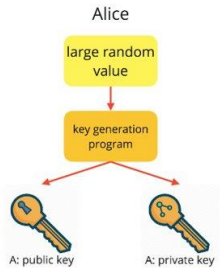
1- Crea un archivo ZIP o RAR protegido con contraseña. Elige una contraseña "fácil".
2- Ubica el archivo en tu máquina Parrot
3- Utiliza John the Ripper para crackear la contraseña del archivo

Realiza los mismos pasos utilizando una contraseña más "segura". ¿Qué diferencias existen con respecto al otro caso?

- Trata de usar un diccionario grande como rockyou.txt (repositorio de DanielMiessler)
- En caso de que no la encuentre trata de crear un diccionario propio que incluya la contraseña

Recursos:
http://www.reydes.com/d/?q=Romper_la_Contrasena_de_un_Archivo_ZIP_utilizando_John_The_Ripper

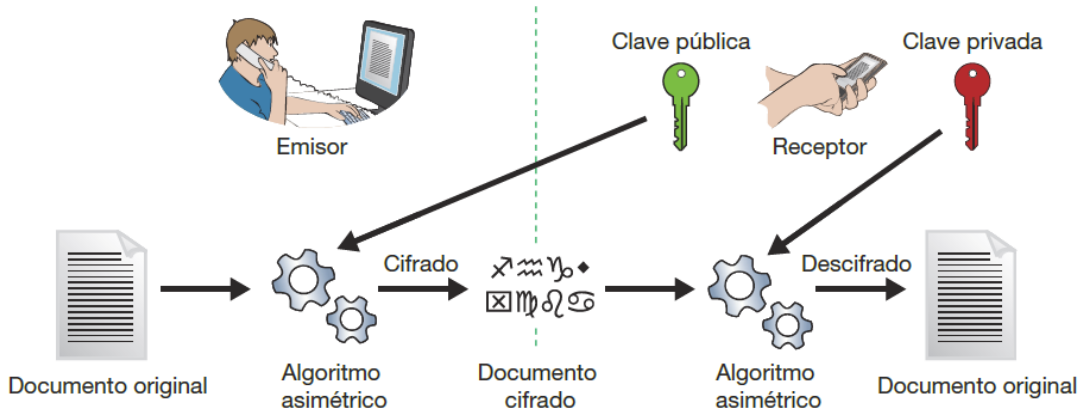
Criptografía



CIFRADO ASIMÉTRICO

En los años setenta, los criptógrafos Diffie y Hellman publicaron sus investigaciones sobre criptografía asimétrica. Su algoritmo de cifrado utiliza dos claves matemáticamente relacionadas de manera que lo que cifras con una solo lo puedes descifrar con la otra.

Cuando el emisor quiere hacer llegar un mensaje confidencial al receptor, primero consigue la clave pública del receptor. Con esa clave y el documento original, aplica el algoritmo asimétrico. El resultado es un documento cifrado que puede enviar al receptor por cualquier canal. Cuando el mensaje cifrado llega al receptor, él recupera el documento original aplicando el algoritmo asimétrico con su clave privada.
Si el receptor quiere enviar al emisor una respuesta cifrada, debería conseguir la clave pública del emisor y seguir el mismo procedimiento



Criptografía

CIFRADO ASIMÉTRICO

La criptografía asimétrica resuelve los dos problemas de la clave simétrica:

- No necesitamos canales seguros para comunicar la clave que utilizaremos en el cifrado. Podemos adjuntarla en nuestros correos, añadirla al perfil de nuestras redes sociales, «postearla» en un blog, incluso repartirla en octavillas por la calle.

- No hay desbordamiento en el tratamiento de claves y canales. Si somos nueve empleados, solo necesitamos nueve claves y un solo canal: la intranet de la empresa, un correo destinado a toda la empresa, etc. Y si aparece un empleado nuevo, serán diez claves y el mismo canal.

CIFRADO ASIMÉTRICO

Sin embargo, los algoritmos asimétricos tienen sus propios problemas:

- Son poco eficientes
- Utilizar las claves privadas repetidamente es arriesgado porque algunos ataques criptográficos se basan en analizar paquetes cifrados
- Hay que proteger la clave privada (especialmente en cuanto a la confidencialidad y la disponibilidad)
- Hay que transportar la clave privada para poder usarla. No es como una clave simétrica, que puede ser recordada para su uso, sino que son cientos de símbolos, sin sentido desde el punto de vista humano.

Criptografía

TRABAJO DE CLASE

Comparte con un compañero un mensaje cifrado utilizando el cifrado asimétrico con GPG.

Recursos:
Cifrado asimétrico con GPG en Linux - Tutorial con ejemplos - Parzibyte's blog

Criptografía

TRABAJO DE CLASE

Asentando conceptos: Cifrados simétrico y asimétrico.

Investiga sobre:

- *¿Qué es el protocolo SSH?*
- *¿Para qué se usa?*
- *¿Qué tipo de cifrados utiliza? Describe de manera general cómo se utilizan los cifrados en una conexión normal mediante este protocolo*

Criptografía

PROTOCOLO SSH

SSH (Secure SHell) es un protocolo de comunicaciones que permite cifrar la conversación extremo a extremo. Se utiliza para sesiones interactivas de comandos (es un buen sustituto de telnet), transferencias de archivos (sustituto de FTP), túneles seguros entre aplicaciones, etc.

Dado que el cifrado asimétrico tiene bajo rendimiento este protocolo (entre otros) utiliza un esquema híbrido:

- Criptografía asimétrica solo para el inicio de la sesión, usado para acordar la clave simétrica aleatoria que se utilizará
- Criptografía simétrica durante el resto de la transmisión. Generalmente se suele cambiar la clave simétrica cada cierto tiempo (minutos) para dificultar más el espionaje de la conversación

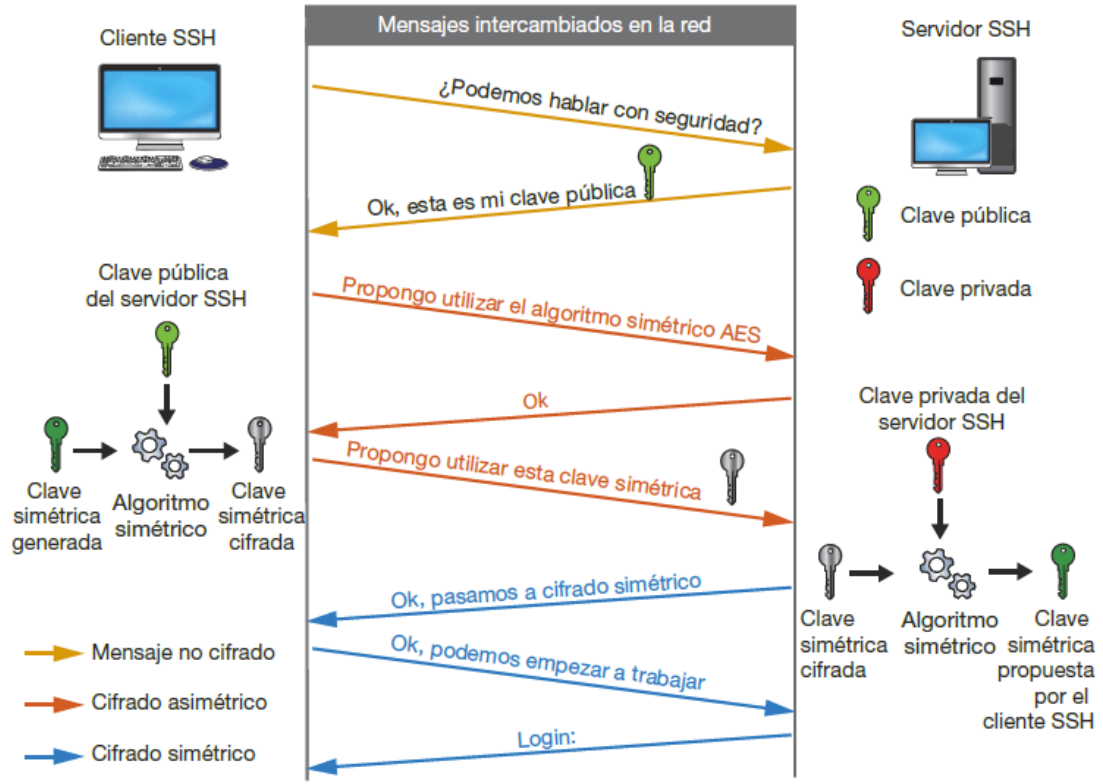


Fig. 2.37. *Esquema híbrido de cifrado en SSH.*

Criptografía

EL PROTOCOLO HTTPS

Uno de los protocolos que tratan de aumentar nuestra seguridad cuando navegamos por Internet es el procolo HTTPS.

HTTPS (Hyper Text Transfer Protocol Secure). En sí mismo HTTPS no es más que HTTP normal sobre SSL/TLS.

SSL/TLS (Secure Sockets Layer/Transmission Layer Security) son dos protocolos para enviar paquetes cifrados a través de Internet, siendo el último el más moderno. Sirven igual para HTTP que para cualquier otro protocolo de comunicación

Cualquier conexión establecida con HTPPS sigue los siguientes pasos:

1. Se acuerdan detalles técnicos entre navegador y servidor (versión del protocolo, algoritmos de cifrado asimétrico y simétrico que se usarán...)
2. el navegador cifra una preclave generada en el momento con la clave pública del servidor al que nos queremos conectar. Eso se envía al servidor, que descifra la preclave con su clave privada.
3. Tanto el servidor como el navegador aplicarán un cierto algoritmo a la preclave y *obtendrán la misma clave de cifrado*

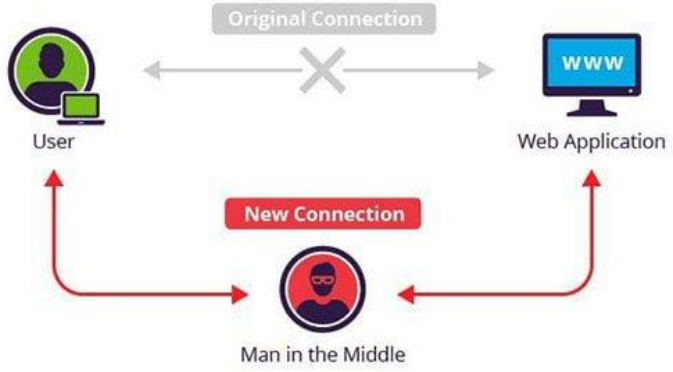
Criptografía

EL PROTOCOLO HTTPS

HTTPS no sólo impide que alguien vea las páginas web que estamos visitando. También impide que puedan conocer las URLs por las que nos movemos, los parámetros que enviamos al servidor (por ejemplo, los usuarios y contraseñas se envían como parámetros POST normalmente) o las cookies que enviamos y recibimos

¿Cuál es la vulnerabilidad más clara de HTTPS?

Que alguien se haga pasar por el servidor con el que quiero conectarme. Normalmente a través de un ataque Man In The Middle (MITM)

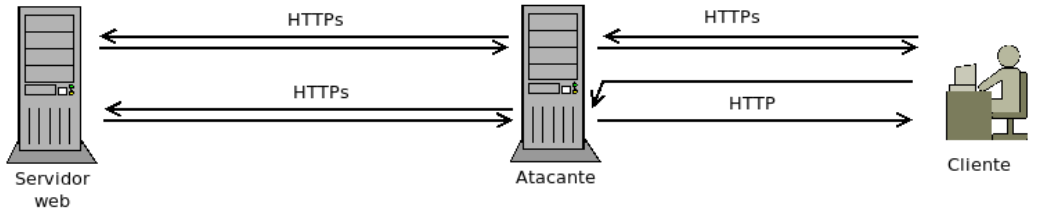


Criptografía

EL PROTOCOLO HTTPS

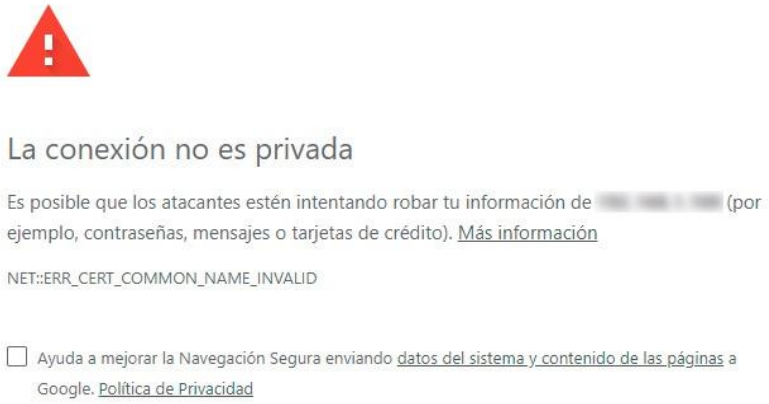
Una vez un atacante hace un MITM entre la víctima y el servidor al que se conecta puede:

- Mantener con el usuario una conversación bajo HTTP y mandar al servidor las peticiones en HTTPS
- Establecer dos conexiones HTTPS distintas: una con la victima y otra con el servidor, haciendo de intermediario y descifrando y cifrando el mensaje antes de reenviarlo hasta su destino.



En ambos casos la responsabilidad de saber si la conexión es segura **recae en el usuario**:

- Debería comprobar que la conexión sea en HTTPS (y no en HTTP)
- Debería comprobar la autenticidad del servidor (atento a los mensajes de alerta sobre incidencias con los certificados del servidor)



Criptografía

AUTENTICACIÓN: CIFRADO Y FIRMA

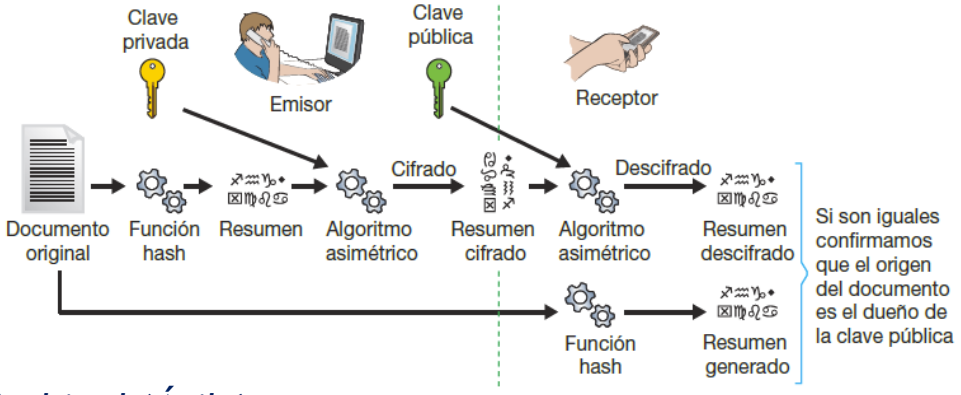
La criptografía, tiene otra aplicación fundamental: determinar la **autenticidad** del emisor de un dato
Esto es factible gracias a la **criptografía asimétrica:**

1.El emisor aplica al documento una función resumen (función hash). El resultado de esta función es una lista de caracteres, que la función garantiza que solo se pueden haber obtenido con el documento original

2. Ahora el emisor cifra ese resumen con su clave privada y lo envía al destino, junto con el documento original.

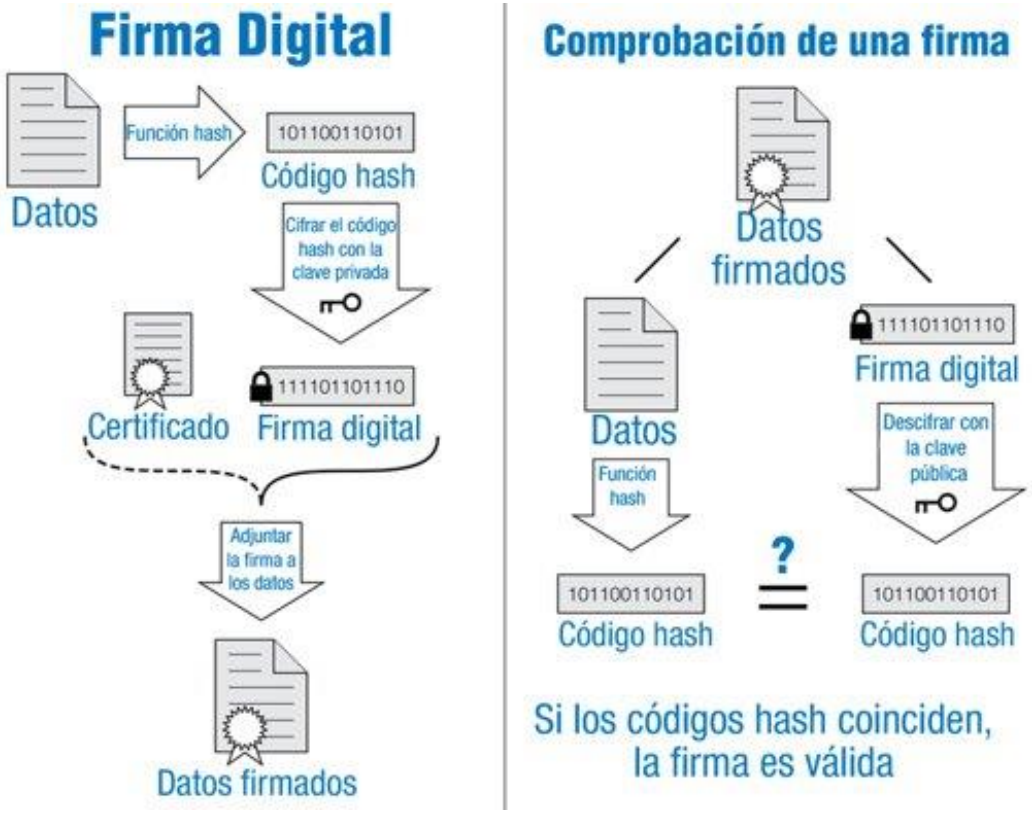
3. En el destino se hacen dos operaciones:
- Aplicar la misma función hash al documento para obtener su resumen.
- Descifrar el resumen recibido, utilizando la clave pública del emisor.

4. Si ambos resúmenes coinciden, el destino puede estar seguro de que el emisor del documento es el mismo que el dueño de la clave pública que acaba de aplicar para descifrar el resumen recibido



Criptografía

FIRMA DIGITAL



AUTENTICACIÓN: CIFRADO Y FIRMA

Por supuesto, si queremos que el documento original no pueda ser interceptado en la transmisión desde el emisor al receptor, debemos cifrarlo. Para ello usaremos la clave pública del receptor. El procedimiento completo sería:

1. El emisor aplica la función hash al original para generar el resumen.
2. El emisor toma su clave privada para aplicar el algoritmo asimétrico al documento resumen. El resultado es un documento resumen cifrado.
3. El emisor toma la clave pública del receptor para aplicar el algoritmo asimétrico al documento original y al documento resumen. El resultado es un documento conjunto cifrado que se envía al receptor.
4. En el receptor, utiliza su clave privada para descifrar los documentos y la clave pública del origen para comprobar la firma

Criptografía

PKI

Cuando trabajamos con claves asimétricas encontramos un problema recurrente: se necesita comparar la huella de la clave importada con la huella de la clave original, para **estar seguros de que vamos a comunicarnos con la persona correcta**.

Cuando nos comunicamos con lugares remotos, no podemos entrar en sus máquinas para ver las huellas ni negociar con cada uno otro canal seguro donde poder consultarlas

La solución a este problema es la implantación de una **PKI (Public Key Infrastructure, infraestructura de clave pública).** Ahora, en la comunicación segura entre cliente y servidor aparecen nuevos interlocutores:

- La **Autoridad de Certificación** (CA [Certificate Authority]), cuya misión es emitir certificados. Hasta ahora los generábamos nosotros mismos con una herramienta en el ordenador.
- La **Autoridad de Registro** (RA [Registration Authority]), que es la responsable de asegurar que el solicitante del certificado es quien dice ser. Por ejemplo, en los certificados necesarios para presentar la declaración de la renta, la solicitud se puede hacer por Internet, pero para recogerlos hay que presentarse con el DNI en una oficina de la Administración.
- La **Autoridad de Validación** (VA [Validation Authority]) es la responsable de comprobar la validez de los certificados digitales emitidos. En la práctica suele coincidir con la CA.
- Los **repositorios**. Son almacenes de certificados. Los principales son el repositorio de certificados activos y el repositorio de listas de revocación de certificados (certificados que, por cualquier motivo, fueron expresamente desactivados antes de caducar).

Referencias

Seguridad informática. **José Fabián Roa Buendía**. **McGraw-Hill** España, 2013.

Seguridad informática (Edición 2020) **POSTIGO PALACIOS**, **ANTONIO**. Ediciones Paraninfo, S.A., May 19, 2020