UCA Universidad de Cádiz
Escuela Superior de Ingeniería

Tema 4
Servidor SSH

servidor que permite acceder remotamente y de forma segura a otros ordenadores

Introducción
Seguridad en Internet
Qué es un servidor SSH
OpenSSH: Instalación y configuración
Métodos de autentificación
Tunneling

Tema 4
Servidor SSH

INTRODUCCIÓN

Introducción

- Seguridad
  - Comprende toda la problemática relacionada con hacer que la información no pueda ser **leída** ni **modificada** por ninguna persona no autorizada
  - La seguridad se aplica
    - A nivel de datos (pérdida, modificación)
    - A intrusos, para evitar acciones maliciosas

privacidad e integridad

proteger el acceso a un fichero determinado
proteger las comunicaciones entre sistemas

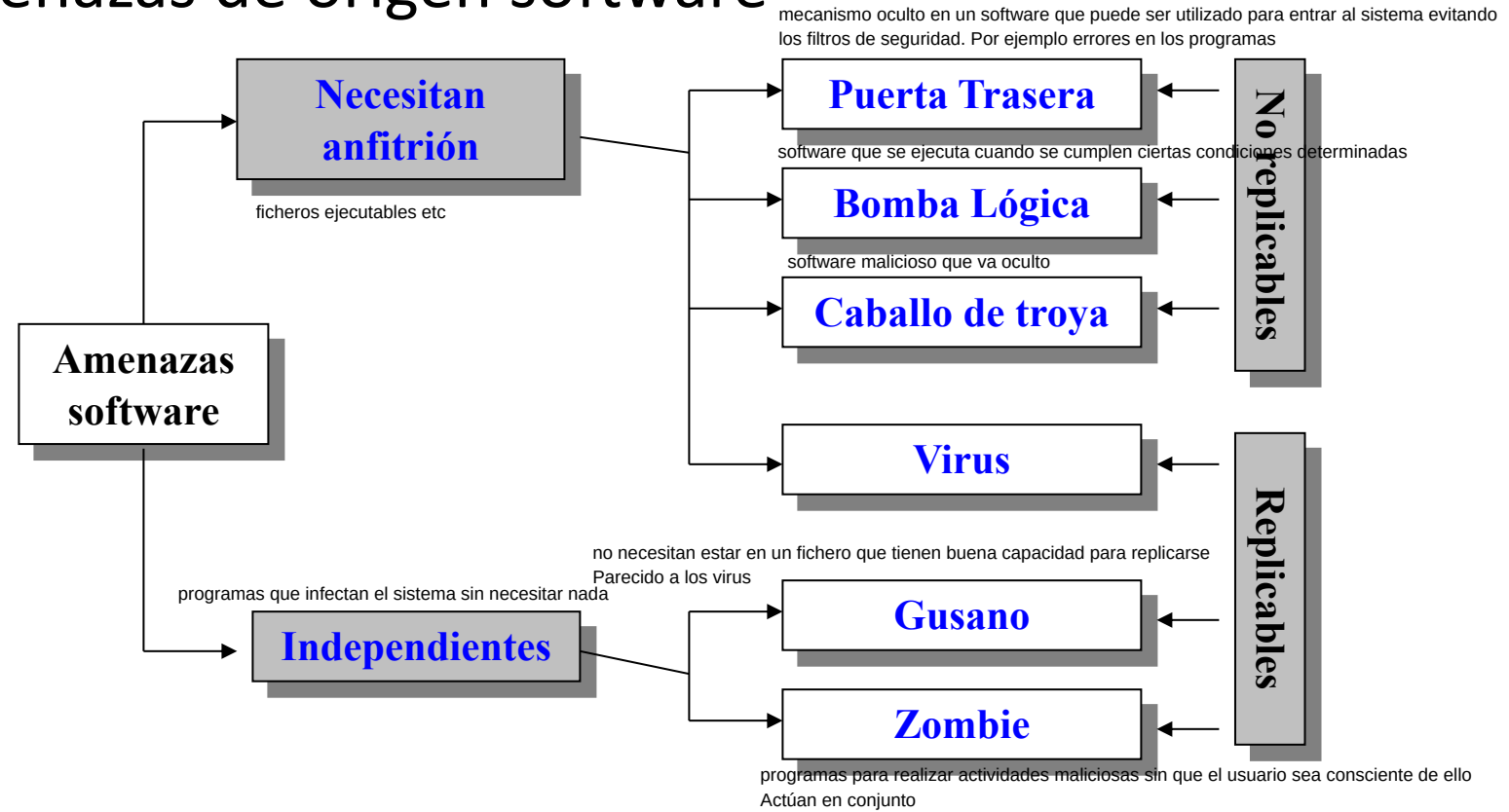
- Las amenazas a la seguridad son de dos tipos básicos
  - Usuarios (hackers)
  - Amenazas de origen software

personas

programas

Introducción

- Amenazas de origen software



Control de acceso

- **Contraseña**
  - Clásico
  - El más utilizado

- Los sistemas de posesión de un **objeto físico** son los más antiguos en control de accesos físicos (llaves, sellos, salvoconductos) está en desuso
  - Hasta hace poco no han sido muy usados

- Sistemas **biométricos** basados en IA, no siempre son seguros
  - Problemas de precio, costumbre de uso, etc.

- Autenticación de **doble factor**

- Todos los sistemas **se pueden combinar** para aumentar la seguridad

UCA Universidad de Cádiz
Escuela Superior de Ingeniería

Tema 4

Servidor SSH

SEGURIDAD EN INTERNET

Criptografía

operaciones matemáticas que transforman un mensaje legible en uno inteligible

UCA Universidad de Cádiz
Escuela Superior de Ingeniería

- Ciencia que utiliza las matemáticas para cifrar y descrifrar datos
  - Herramienta para **proporcionar servicios de seguridad** en los sistemas informáticos

- Definiciones
  - **Cifrado**
    - Proceso de disfrazar un texto legible (texto, fichero, imagen, sonido, ...) para convertirlo en ininteligible
  - **Descrifrado**
    - Conversión de un texto cifrado a texto legible

- Características
  - Algoritmos conocidos — los algoritmos para cifrar la info debe ser algoritmos conocidos
  - Operación eficiente — cifrar y descifrar ha de ser operaciones eficientes con un tamaño aceptable
  - Pequeño tamaño de la información cifrada

Algoritmo criptográfico

- **Función de cifrado** *E* sobre el mensaje M produce el texto cifrado (o criptograma) C:

$$E(M) = C$$

no valido porque todo el mundo sabe como cifrar y descifrar

- **Función de descifrado** D sobre C produce M:

$$D(C) = M$$

- Algoritmo criptográfico o de cifrado:
  - Función utilizada para cifrar y descifrar

Algoritmos criptográficos basados en claves

- Son los más comunes
  - Los algoritmos son **públicos**
  - La(s) clave(s) determina(n) el resultado de los procesos de cifrado/descifrado
    - Cifrado: *E(M, K) = C* a la operación de cifrado le añadimos una clave
    - *Descifrado: D(C, K') = M* con otra clave podemos descifrar el criptograma y obtener el mensaje original

- La seguridad de estos algoritmos reside en **mantener secreta(s) la(s) clave(s)**

- **Calidad** del algoritmo
  - Dificultad para adivinar la clave
  - Velocidad de los procesos de cifrado/descifrado

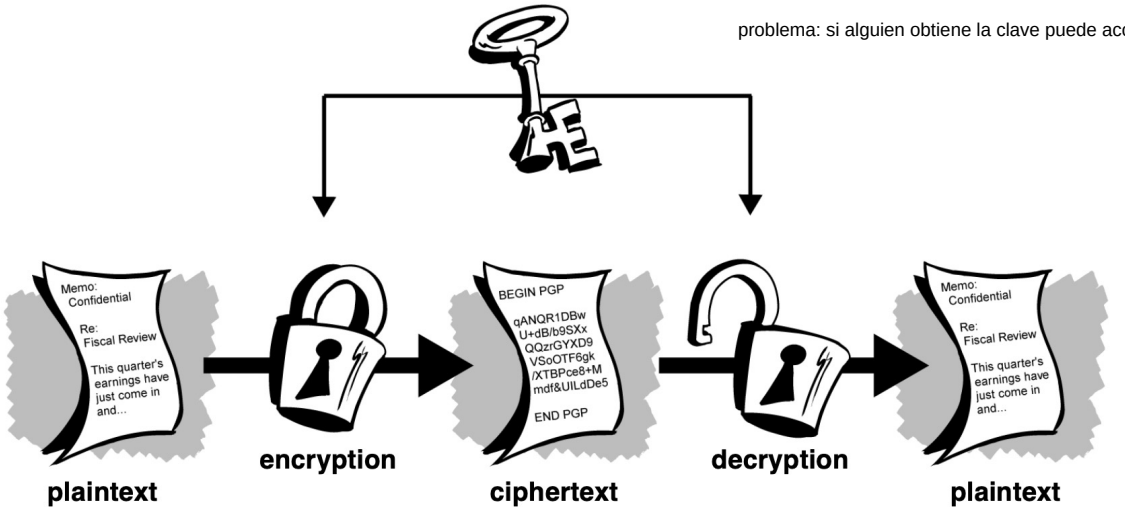
Algoritmos simétricos

comunicaciones cifradas entre dos servidores

- Cifrado clásico: Clave K para las operaciones de cifrado y de descifrado

$$E_K(M) = C \quad D_K(C) = M \quad D_K(E_K(M)) = M$$

problema: si alguien obtiene la clave puede acceder a la información



Introduction to Cryptography, Network Associates Inc

Algoritmos simétricos

ya no se usa

- También llamados algoritmos de **clave secreta**
  - La clave para cifrar **se puede calcular** a partir de la clave para descifrar y viceversa (coinciden en la mayoría de los casos)
- Requieren que el emisor y el receptor acuerden, a priori, una clave
- Ejemplos de criptosistemas simétricos
  - Data Encryption Standard (DES)
    - DES
    - DES triple
    - GDES
  - NEWDES
  - FEAL (*Fast Encryption Algorithm*)
  - IDEA (*International Data Encryption Standard*)

Algoritmos simétricos

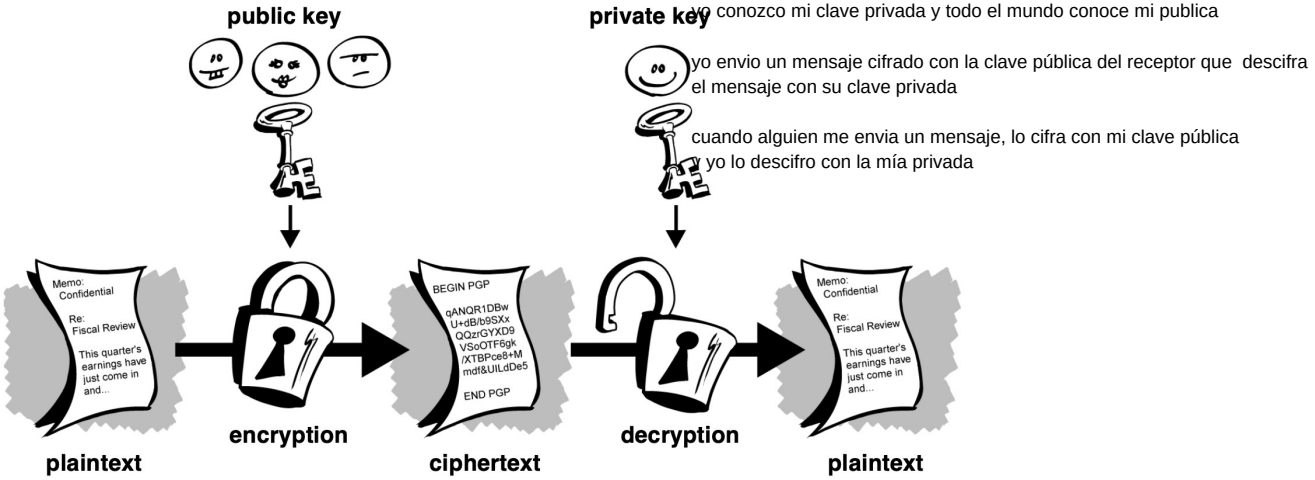
- Ventajas
  - **Rápidos** en los procesos de cifrado y descifrado
  - Posibilidad de **claves sencillas y cortas**, establecidas por el usuario

- Inconvenientes
  - **Acuerdo de clave** entre emisor y receptor
  - **Distribución segura** de las claves
  - **Salvaguarda** de las claves
  - Si una clave se descubre:
    - Se pueden **descifrar** mensajes
    - Se pueden **producir** mensaje nuevos

Algoritmos asimétricos

- **Cifrado de 2 claves (pública y privada)**: claves diferentes para cifrar descifrar (K_E y K_D)

$$E_{KE}(M) = C \qquad D_{KD}(C) = M$$



Introduction to Cryptography, Network Associates Inc

Algoritmos asimétricos

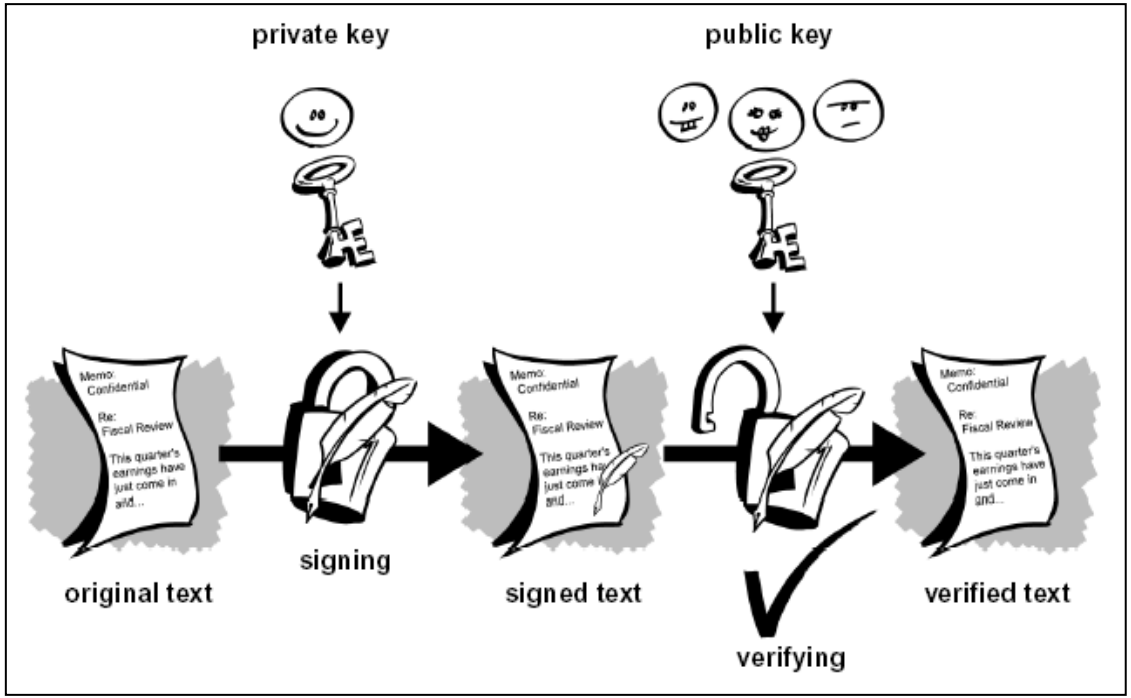
- También llamados algoritmos de **clave pública**
  - Clave de cifrar ≠ clave para descifrar
  - No pueden ser calculada una a partir de la otra

- La clave para **cifrar** se puede hacer **pública** (clave pública)
  - Cualquiera puede cifrar un mensaje
  - Sólo quien tenga la correspondiente clave de descifrado (clave privada) puede descifrar

- Se puede utilizar la clave privada para cifrar y la pública para descifrar; esto se llama **firma digital**

Firma digital

- Firma digital



Introduction to Cryptography, Network Associates Inc

Algoritmos asimétricos

La clave privada no se sube al servidor ssh, subimos la pública

- Ventajas
  - Es “computacionalmente imposible” deducir la clave privada a partir de la clave pública
  - Cifrado: Cualquiera con la clave pública
  - Descifrado: Sólo con la clave privada
    - **La clave pública no puede descifrar**
  - En un sistema de n usuarios se manejan 2n claves [n(n–1)/2 en los simétricos]

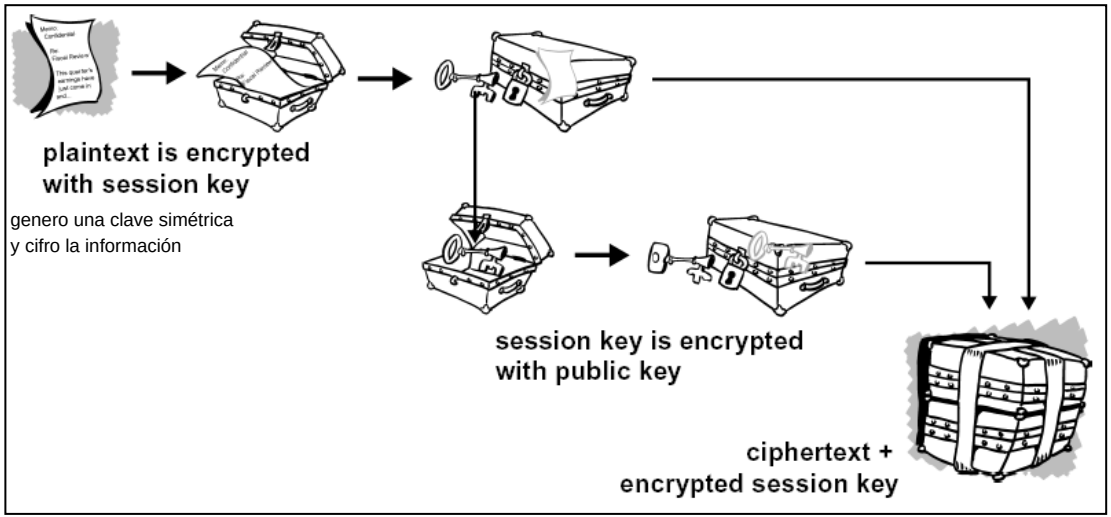
- Inconvenientes
  - Claves de gran longitud
  - ~1000 veces más lento que los simétricos

Sistemas híbridos

utilizo RSA Para enviar una clave de sesión con el ordenador que me quiero conectar
Cuando se envía un ACK se establece la comunicación

- Combinan criptografía de clave privada y pública
- Ejemplo: PGP (Pretty Good Privacy)
- Encriptación:

mando el paquete cifrado con la clave cifrada mediante RSA



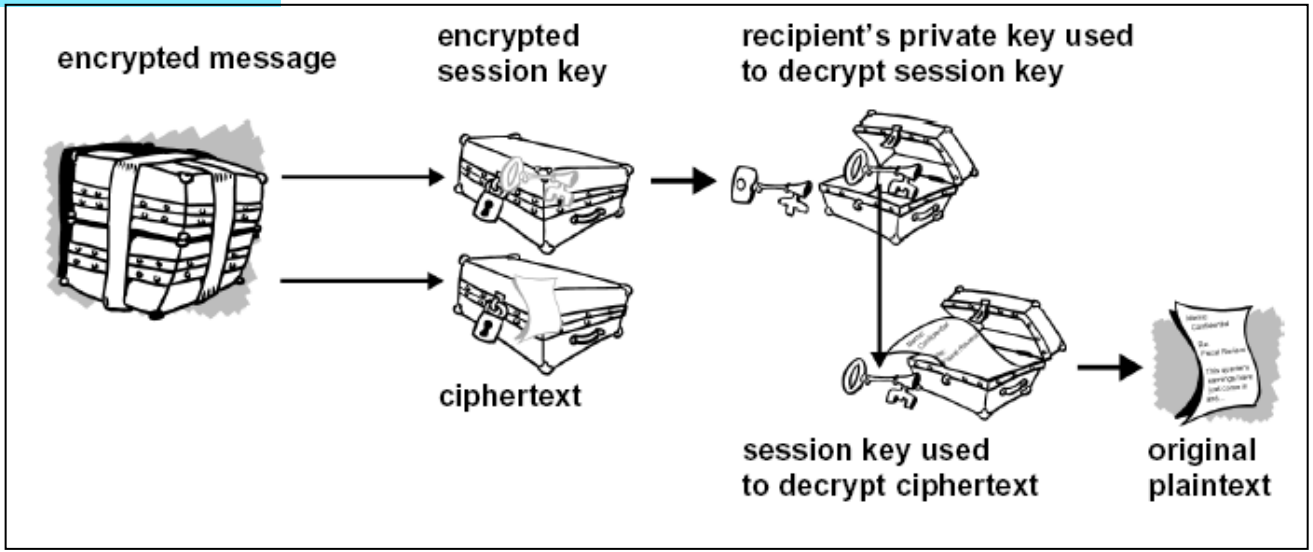
genero una clave simétrica
y cifro la información

Introduction to Cryptography, Network Associates Inc

Sistemas híbridos

- Combinan criptografía de clave privada y pública
- Ejemplo: PGP (Pretty Good Privacy)
- Desencriptación:

descifro el primer paquete con la clave publica del usuario que me ha mandado y tengo la clave



Introduction to Cryptography, Network Associates Inc

Sistemas híbridos

- Ventajas
  - Lo mejor de cada sistema
    - **Velocidad** de los sistemas simétricos
    - **Seguridad** de los asimétricos
  - Resuelve los inconvenientes de cada sistema
    - **Distribución de claves** de los simétricos
    - **Complejidad computacional** de los asimétricos

Certificados digitales

hay una entidad certificadora que asgura que una clave pública pertenece a una persona

- Los certificados digitales
  - Permiten identificarnos en la red Internet
- Nace para resolver el problema de
  - Administrar las claves públicas y que la identidad del dueño no pueda ser falsificada
- Fundamento
  - Una tercera entidad (*Certificate Authority*) interviene en la administración de las claves públicas y asegura que éstas tengan asociado un usuario claramente identificado

El protocolo Secure Socket Layer (SSL)

protocolo para enviar información cifrada

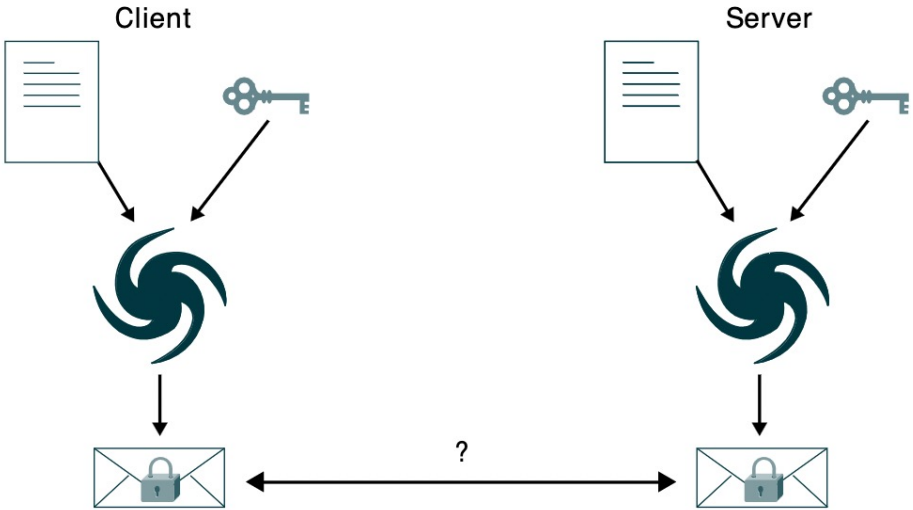
UCA Universidad de Cádiz
Escuela Superior de Ingeniería

- Desarrollado por Netscape para **transmitir documentos** privados por Internet
- Capa de seguridad sobre protocolos de transporte (http, ftp, …)
- **Provee** seguridad sobre el tráfico Web
  - Confidencialidad
  - Integridad de los mensajes
  - Autenticación
- **Elementos** de seguridad
  - Criptografía
  - Firmas digitales
  - Certificados
- Por convención
  - Las URLs que requiren SSL comienzan por `https`, `ftps`, …

El protocolo Secure Socket Layer (SSL)

- Integridad de los **mensajes**
  Cada mensaje lleva una firma digital adjunta
- Integridad del **canal de comunicación**

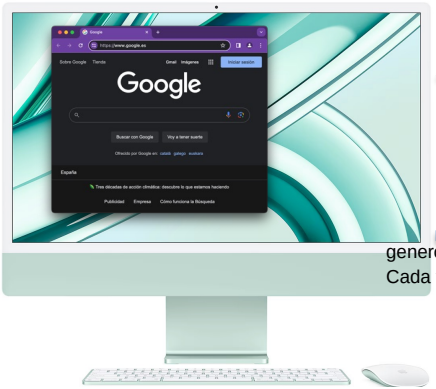
los mensajes se cifran con una clave de sesión
y esa info va cifrada con la clave de la comunicación



Introduction to Secure Sockets Layer, White paper, Cisco Systems

SSL handshake

1. Solicitud de inicio de sesión SSL



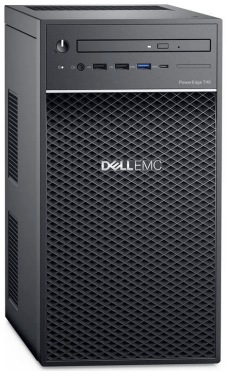
el serveer envía el certificado para poder verificar que es el servidor de google por ejemplo

2. Envío del certificado y clave pública del servidor

3. El cliente comprueba la autenticidad del certificado (consulta a autoridad certificadora)

genero la clave de sesión simétrica y envio la info al servidor con su clave pública y este me genera la clave de sesión.
Cada vez que repito el proceso se genera una clave de sesión distinta

4. Generación y envío de clave simétrica. Cifrada con la clave pública del servidor



5. Canal de comunicación seguro con clave simétrica

El protocolo SSL

- OpenSSL
  - Proyecto open source para implementar SSL
  - http://www.openssl.org
- Versiones
  - SSL 3.0 (1996)
  - TLS 1.0 (1999)
    - Transport Layer Security
  - TLS 1.1 (2006)
  - TLS 1.2 (2008)
  - TLS 1.3 (2018)

SSL – Gestión de claves y certificados

- Para configurar SSL
  1. Generar una **clave privada** fuerte
  2. Obtener un **certificado**
  3. Instalar el certificado en el servidor

SSL – 1. Generación de la clave

metodos de encriptación para SSL

- **Métodos** compatibles: RSA, DSA, ECDSA
  - Decisión en función del uso
    - Servidor web: RSA (DSA limitado a 1024 bits)
    - Servidor SSH: DSA y RSA muy utilizados
      - ECDSA: Soporte de CA limitado, no implementado en muchos clientes

con una clave más pequeña te proporciona el mismo nivel de seguridad que el RSA y el DSA. No se usa porque los organismos de autenticación no son compatibles

SSL – 1. Generación de la clave

- **Tamaño** recomendado
  - DSA y RSA: 2,048 bits
    - Claves de 512 bits se adivinan con fuerza bruta a partir de los mensajes cifrados
  - ECDSA: 256 bits

SSL – 1. Generación de la clave

con el programa OPEN SSL puedes generar la clave.

- Se puede proteger la clave privada con **contraseña**
  - Aumenta el nivel de **seguridad de la clave** privada
    - Permite el almacenamiento, transporte y backup seguros de la clave privada
  - No aumenta el nivel de seguridad del sistema
    - La clave privada está en memoria, por lo que puede extraerse
  - No se pueden usar sin introducir la contraseña

SSL – 1. Generación de la clave

- Ejemplo:
  - Generación de clave privada RSA

```
$ openssl genrsa –aes128 –out id_rsa 2048
```

Tamaño clave privada

Protocolo cifrado contraseña

Fichero clave privada

al servidor SSH le doy la clave pública de la gente que quiero que se conecte

Al copiar y pegar la clave al archivo en el servidor hemos de poner toda la clave junta en una sola linea

  - Generación de clave pública RSA

```
$ openssl rsa -in id_rsa -pubout -out id_rsa.pub
```

Generar clave pública

SSL – 1. Generación de la clave

- **Verificar** ficheros generados

```
$ cat id_rsa

-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,3B3F0E9140C1987300D0FB94F79DD983

yCGJ/5zUIONL2sLTsLWohB1sclOudrcuWesSQfGgug13amBMknMgnBHwGE3HmQ60
VZRIVRBrT/GjFHPBuJSWhaL+OK2vQv4CE9YDLgd4np5QTwJwWjp+ludIzZfoNQZq
…
AypUuLLFDoovpCch0nbckQcPLtyvggZVZsIuBQm8S9lzLA+ri98I8LNJY7UcrKr1
-----END RSA PRIVATE KEY-----
```

SSL – 1. Generación de la clave

- **Verificar** ficheros generados

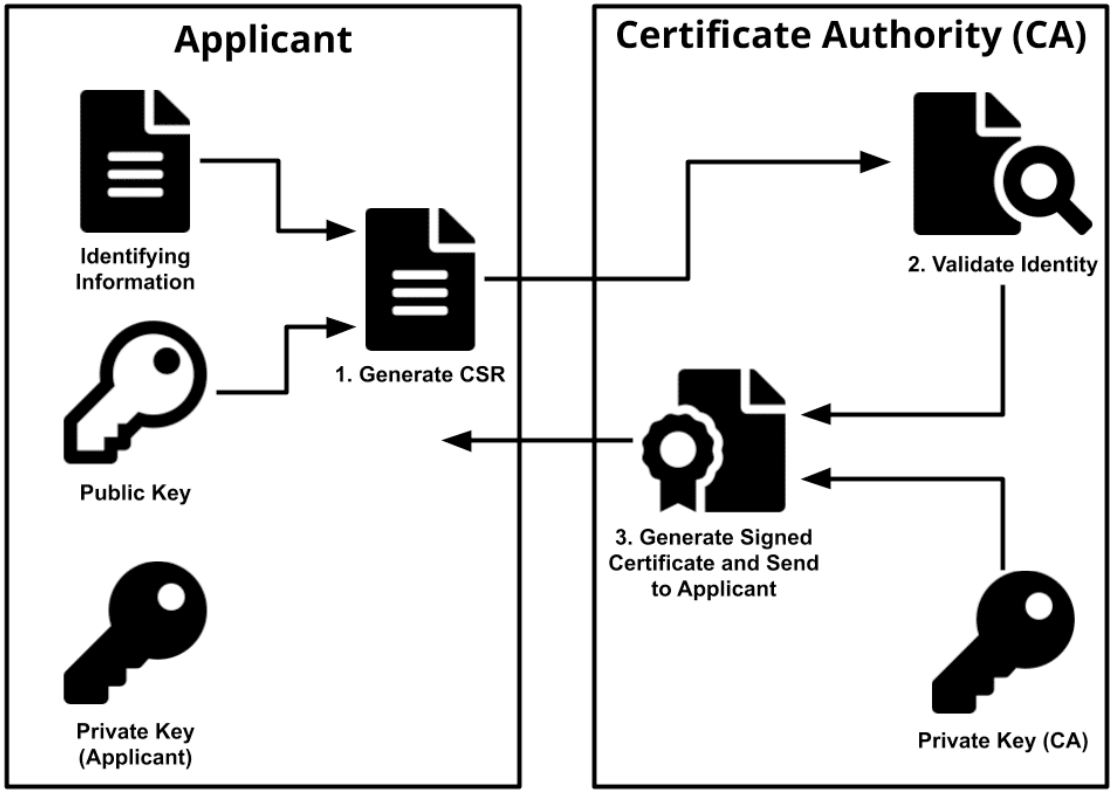
```
$ cat id_rsa.pub

-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAy6R1lEwIA7EWdXMlUcUT
BhroRt6xlCxey0vDYXdlLJrgmjs8ZI9dM4XTqJIKbGIvfQ1WKMF0YHpgpmZPKG1b
DYxIAWTRTQ5XFszZ+ICLIOXa8byqsPY7fwwlvcUIw/ICzbDqn8Wfk60pF6+Yc6Kl
YzLbNsXpz5PtCsPjN4m73jf8o6cFjDk5tOm8uTXjlEDFX1v0sHsKfmoX6qO+9s93
jq6dNJIkrSgaDXghpNXwXE8dnYm9nnufM7C8ZXd19RqDh7+JhwMiyTvwApF3yp1x
mYF2SEpVhad6/2LMS0I1frlX7t/sQ+S6sMgpSabW+/vIsz9afRTJTDhzebwHhyOC
zQIDAQAB
-----END PUBLIC KEY-----
```

SSL – 2. Obtener un certificado

- Paso 1: Crear una solicitud de firma de certificado
- Paso 2: Solicitud del certificado

El protocolo SSL – Autoridad de certificación



https://www.ssl.com

SSL – 2. Obtener un certificado

- Paso 1: Crear una solicitud de firma de certificado
  - Crear una clave privada para el certificado

```
$ openssl genrsa –aes128 –out id_rsa_csr 2048
```

  - Crear la solicitud del certificado

```
$ openssl req -new -key id_rsa_csr -out id_rsa.csr
```

  - Comprobar la solicitud del certificado

```
$ cat id_rsa.csr
```

SSL – 2. Obtener un certificado

- Paso 2: Solicitud del certificado
  - Dos opciones
    - **Obtener un certificado** de una entidad autorizada (CA o *Certificate Authority*). Ejemplo: Verisign
      - Cuesta dinero
      - Necesario si el sitio Web es público y se pueden realizar transacciones
    - Actuar **uno mismo** como entidad autorizada
      - Válido para entornos controlados, como intranets

```
$ openssl x509 -req -days 365 -in id_rsa.csr -signkey fd.key -out id_rsa.crt
```

SSL – 2. Obtener un certificado

- Certificado sólo para un *hostname*
- Para incluir nuevos:

1. Crear un fichero (ej: hosts.ext) con los hostnames:

```
subjectAltName = DNS:*.as.com, DNS:as.com, IP:192.0.2.1, email:user@example.com
```

2. Crear el certificado usando ese fichero:

```
$ openssl x509 -req -days 365 -in id_rsa.csr -signkey id_rsa.key
    -out id_rsa.crt –extfile hosts.ext
```

SSL – 3. Instalar el certificado

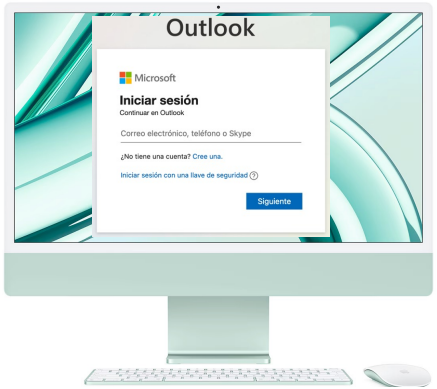
- Copiarlo al directorio donde el servidor lo usará. En el caso del servidor web Apache

  /etc/httpd/conf/ssl.crt

Seguridad en Internet

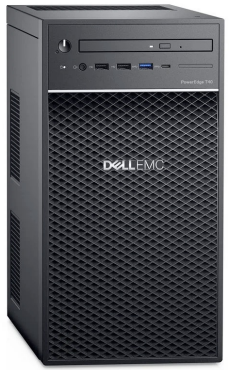
- Comunicaciones seguras:

SSL + autenticación doble factor



Usuario
Contraseña
Código TFA

CIFRADO



Cookie de sesión

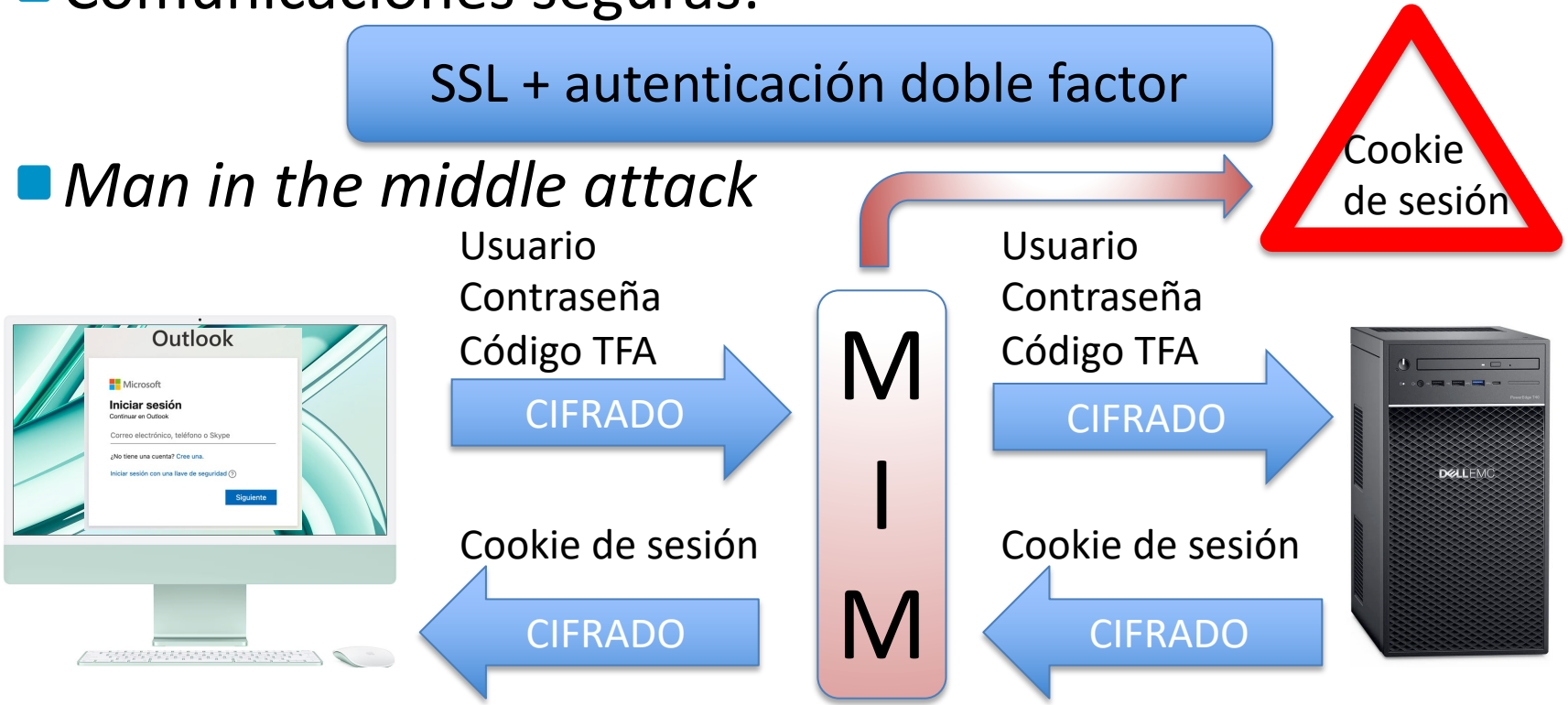
CIFRADO

Seguridad en Internet

- Comunicaciones seguras:

SSL + autenticación doble factor

- *Man in the middle attack*



Tema 4
Servidor SSH

QUÉ ES UN SERVIDOR SSH

Introducción a SSH

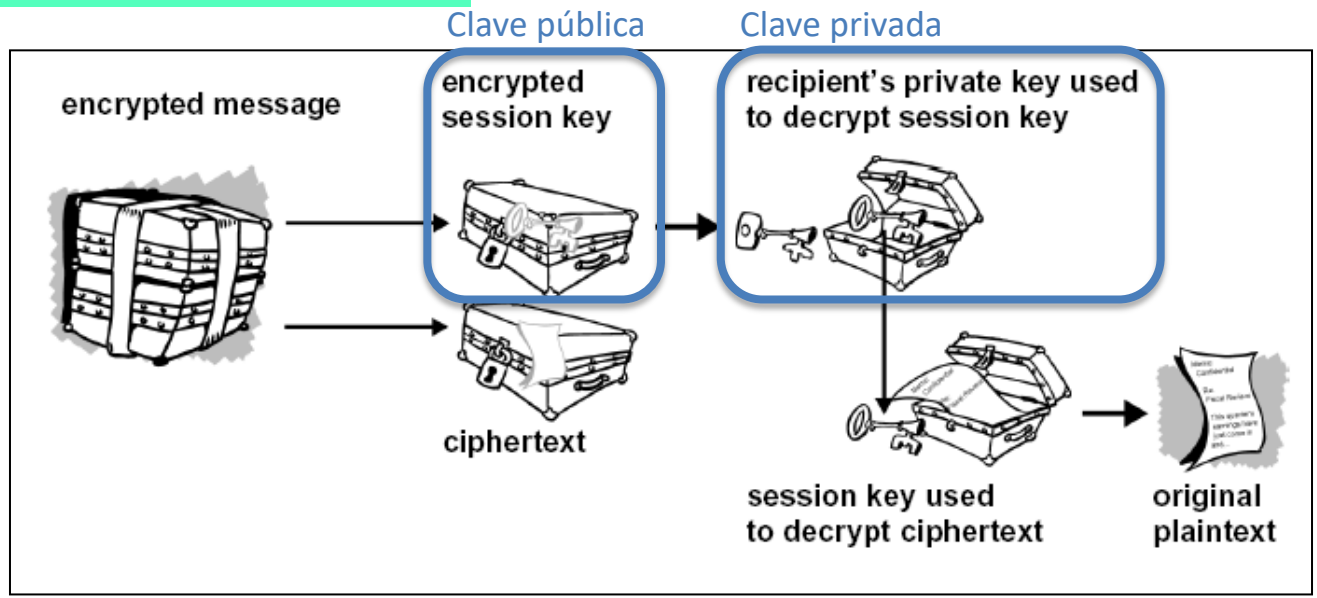
- SSH significa Secure Shell
- Protocolo para compartir datos entre dos ordenadores a través de Internet de forma segura
- Permite establecer una conexión remota segura y encriptada con un servidor
- Múltiples usos:
  - Administrar servidores
  - Transferir archivos herramienta scp para transferir archivos
  - Acceder a bases de datos
  - Controlar versiones

con scp necesitas tener un usuario en el ordenador remoto

Funcionamiento de SSH

hoy en día se usa el híbrido

1. Utiliza un sistema de cifrado asimétrico
   - Clave pública (**compartida**): cifra la clave de sesión
   - Clave privada (**local**): descifra la clave de sesión

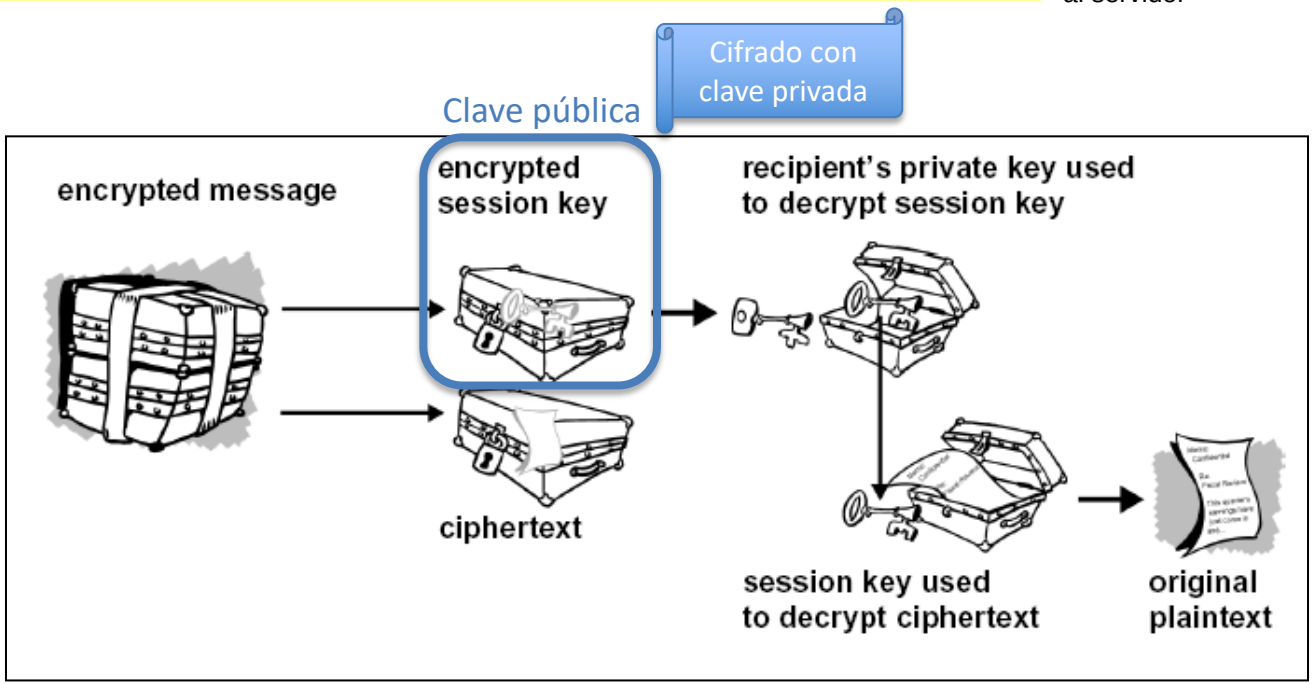


Introduction to Cryptography, Network Associates Inc

Funcionamiento de SSH

2. Utiliza un mecanismo de autenticación, que verifica la identidad del servidor y del cliente

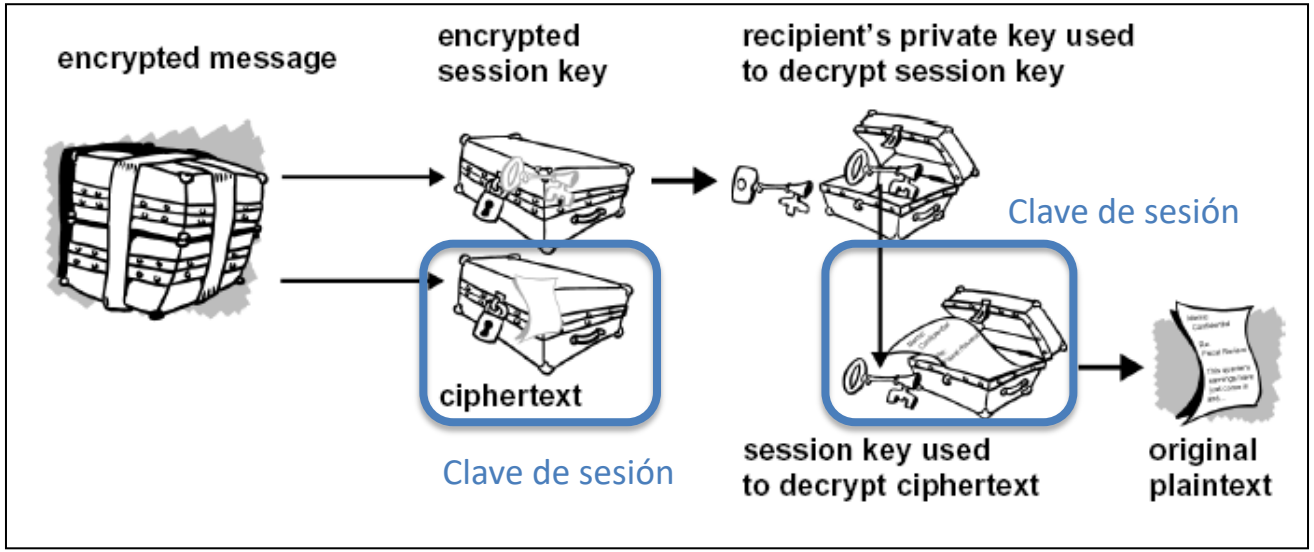
la clave pública permite identificar al servidor



Introduction to Cryptography, Network Associates Inc

Funcionamiento de SSH

3. Utiliza un mecanismo de integridad, que verifica que los datos no hayan sido alterados durante la transmisión



Introduction to Cryptography, Network Associates Inc

Uso de SSH

uso la Kpub del servidor para establecer la conexión

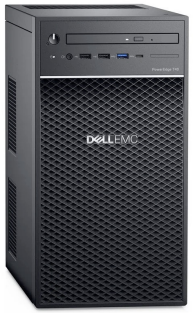
luego se inicia sesión con shh

- Requiere
  - Cliente SSH: programa que se ejecuta en el ordenador del usuario
  - Servidor SSH: programa que se ejecuta en el ordenador remoto

- Clientes SSH existentes: PuTTY, OpenSSH, WinSCP, …

openSSH por defecto en Linux

- Inicio de sesión SSH
  - Nombre de usuario
  - Dirección (IP, URL)
  - Puerto del servidor. **Por defecto: 22**



- Opciones de autenticación **con el servidor**:
  - Clave pública
  - Contraseña de usuario

- Una vez establecida la sesión SSH, se puede acceder a la línea de comandos del servidor y ejecutar las operaciones deseadas

Ventajas de SSH

- **Seguridad**: protege los datos de posibles ataques
  - Espionaje
  - Robo
  - Suplantación
  - Manipulación
- **Versatilidad**: puede usarse para diferentes propósitos
  - Administración de servidores
  - Transferencia de archivos
  - Acceso a bases de datos
  - Control de versiones
- **Eficiencia**: Reduce el tiempo y el esfuerzo necesarios para realizar tareas en servidores remotos, ya que se puede hacer todo desde una sola interfaz

Desventajas de SSH

- **Complejidad**: requiere un cierto nivel de conocimiento técnico para configurarlo y usarlo correctamente

- **Riesgo**: Puede ser vulnerable a algunos ataques
  - Robo o pérdida de claves
  - Reenvío de puertos
  - Agujeros de seguridad

https://www.incibe.es/

- **Limitación**: Sólo se puede usar con sistemas operativos compatibles, como Linux, Unix, Windows, Mac OS, etc.

SSH - Conclusiones

- Protocolo para **compartir datos** entre dos ordenadores a través de Internet **de forma segura** y encriptada

- Se usa para establecer una conexión remota con un servidor y realizar **diferentes operaciones**

- Muchas **ventajas**, como la seguridad, la versatilidad y la eficiencia, pero también algunas **desventajas**, como la complejidad, el riesgo y la limitación

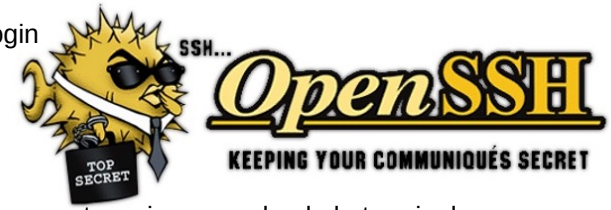
UCA Universidad de Cádiz
Escuela Superior de Ingeniería

Tema 4
Servidor SSH

OPENSSH: INSTALACIÓN Y CONFIGURACIÓN

Open SSH

- Versión **libre** de la familia de protocolos SSH
- OpenSSH incluye
  - **ssh**: reemplaza a telnet y rlogin para login
  - **sftp**: reemplaza a ftp para transferencia de archivos
  - **scp**: reemplaza a rcp para copiar ficheros de local a remoto y viceversa desde la terminal
  - **sshd**: servidor servidor ssh



- Herramientas para **gestión de claves**: ssh-add, ssh-keysign, ssh-keyscan y ssh-keygen

añadir una kpriv al agente ssh que se encarga de gestioanr las claves

genera par de claves pub y priv. Se puede hacer con openssl como hicimos anteriormente

Open SSH

- **Autenticación**: par clave pública/privada
  - Contraseñas “no triviales”
  - Las claves públicas se distribuyen
    - Clientes SSH remotos
    - Servidores SSH remotos

- Toda sesión SSH se inicia con autenticación *host-key*
  - Clave pública del servidor se almacena en el cliente en ~/.ssh
  - Cuenta de usuario en el equipo remoto

Open SSH - Instalación

- Compilación de openSSH
  - `tar xzvf openssh-9.6.tar.gz`
  - `sd ssh`
  - `make obj`
  - `make cleandir`
  - `make depend`
  - `make`
  - `make install`

```
$ apt install openssh-client
$ apt install openssh-server
```

- Posibles dependencias
  - zlib
  - openSSL

Configuración y uso – PRECAUCIONES

¡¡¡Siempre hacer copia del fichero de configuración antes de tocar!!!

Comprueba la nueva configuración antes de reiniciar el servidor:

```
$ sudo sshd -t -f /etc/ssh/sshd_config
```

¡¡¡Siempre tener openSSH actualizado!!!

Configuración y uso

- Fichero de configuración: sshd_config
  - Configuración de un servidor SSH seguro
    - Port
    - PermitRootLogin lo ponemos un no para añadir una capa de seguridad adicional
    - PasswordAuthentication
    - PubkeyAuthentication
    - MaxAuthTries
    - MaxStartups: número máximo de sesiones desde una Ip
    - AllowUsers / DenyUsers
    - AllowGroups / DenyGroups
  - Banner info que te sale al hacer el ssh al servidor

```
$ man sshd_config
```

- Inicios de sesión sin contraseña

```
$ ssh-agent $SHELL
$ ssh-add
```

Iniciar/parar/reiniciar servidor

- `$sudo service sshd ACCIÓN`
- `$sudo /etc/init.d/sshd ACCIÓN`
- `$sudo systemctl ACCIÓN sshd.service`
  - Para distribuciones Linux con systemd

ACCIÓN = `start`
`stop`
`restart`
`status`

- En algunas distribuciones el nombre del script es `ssh` en lugar de `sshd`
  - Debian/Ubuntu

UCA Universidad de Cádiz
Escuela Superior de Ingeniería

Tema 4
Servidor SSH

MÉTODOS DE AUTENTIFICACIÓN

Métodos de autenticación

- Usuario y contraseña del sistema
- Clave pública/privada
- Doble autenticación

Autenticación: usuario y contraseña

- Usuario y contraseña del sistema
  - Fichero /etc/ssh/sshd_config
    - PermitRootLogin no
    - PasswordAuthentication yes
    - PubkeyAuthentication no
    - AllowUsers user1, user2, …

Autenticación segura: clave pública/privada

- Acceso seguro mediante autenticación por clave pública
  - Fichero /etc/ssh/sshd_config
    - PermitRootLogin no
    - PasswordAuthentication no
    - PubkeyAuthentication yes
    - AllowUsers user1, user2, …

- Generación de claves
  - `ssh-keygen -t dsa -f ~/.ssh/id_dsa`
  - `ssh-keygen -t rsa -b 4096 -f ~/.ssh/id_rsa`
  - **Copiar clave pública** en la cuenta remota:
    - Automáticamente: `ssh-copy-id usuario@IP_servidor`
    - Manualmente: Añadir la clave pública al fichero `authorized_keys` del usuario en el servidor

Autenticación segura: doble autenticación

- Usuario/password + One Time Password (OTP)

- Instalación
  - Dependencias

```
$ sudo apt install libpam0g-dev make gcc wget ssh
```

  - Google Authenticator

```
$ sudo apt install libpam-google-authenticator
```

Autenticación segura: doble autenticación

- Configuración automática

```
$ google-authenticator
```

¡¡¡En cada usuario!!!

- Fichero de configuración

```
/root/.google_authenticator

" RATE_LIMIT 3 30
" WINDOW_SIZE 17
" DISALLOW_REUSE
" TOTP_AUTH
79114586
69973536
30276544
49485321
37835027
```

3 intentos de login

Código nuevo cada 30 segundos

8 códigos permitidos por delante y detrás

El mismo token no se puede utilizar más de una vez

Autenticación segura: doble autenticación

- Configurar ssh
  - /etc/ssh/sshd_config

```
ChallengeResponseAuthentication yes
```

  - /etc/pam.d/sshd

```
auth required pam_google_authenticator.so
```

UCA Universidad de Cádiz
Escuela Superior de Ingeniería

Tema 4
Servidor SSH

TUNNELING

Tunneling

- Tunneling: encapsulación de un protocolo de red dentro de otro protocolo
- Ejemplos de tunneling
  - **Port forwarding**: Acceder a servicios internos de una red desde fuera
  - **VPN**: permite acceder de forma segura a una red privada a través de una pública
  - **HTTPS**: el navegador establece un túnel seguro para la conexión con el servidor

Tunneling puerto local

- Conexión segura de **puerto local** con uno remoto

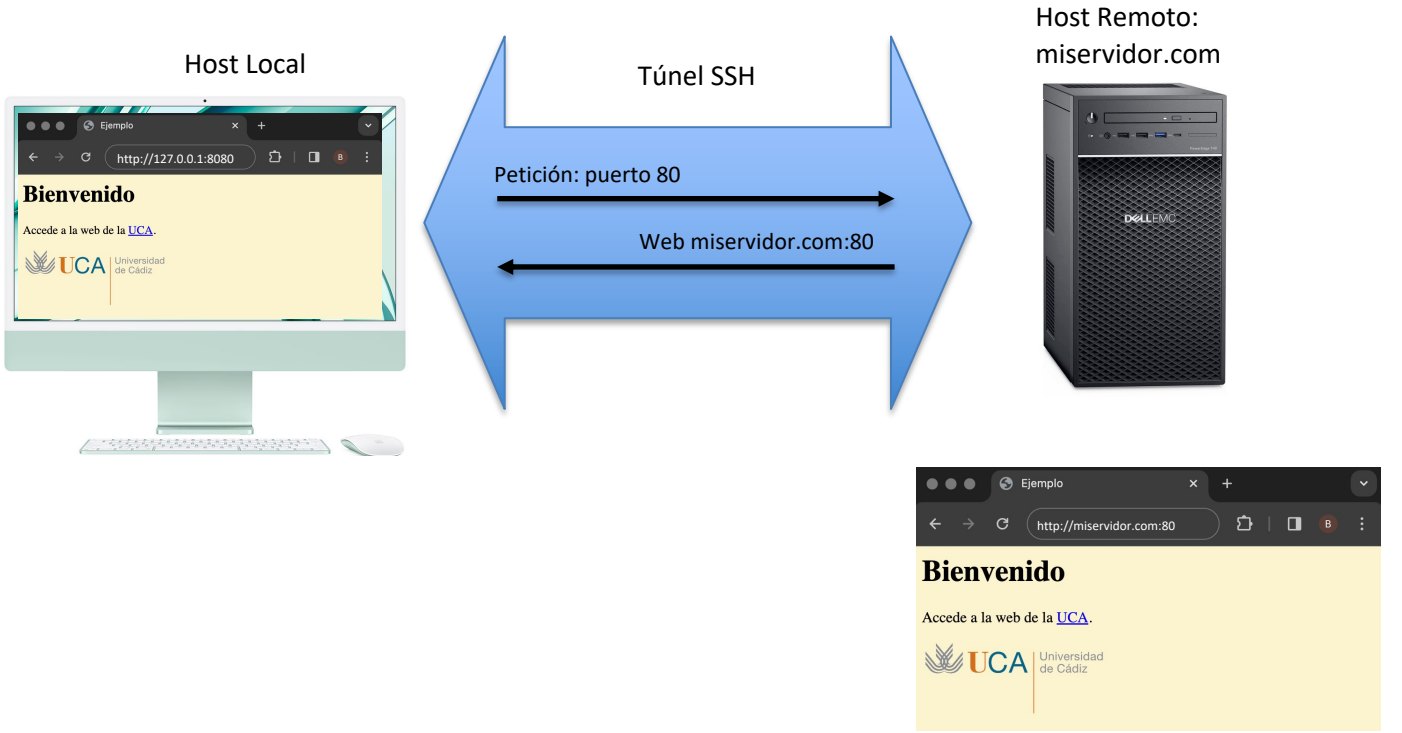
```
$ ssh –L puertoLocal:localhost:puertoRemoto usuario@hostRemoto
```

- Ejemplo

```
$ ssh –L 8080:localhost:80 usuario@miservidor.com
```

Tunneling puerto local

```
$ ssh –L 8080:localhost:80 usuario@miservidor.com
```



Tunneling puerto remoto

- Conexión segura de **puerto remoto** con uno local

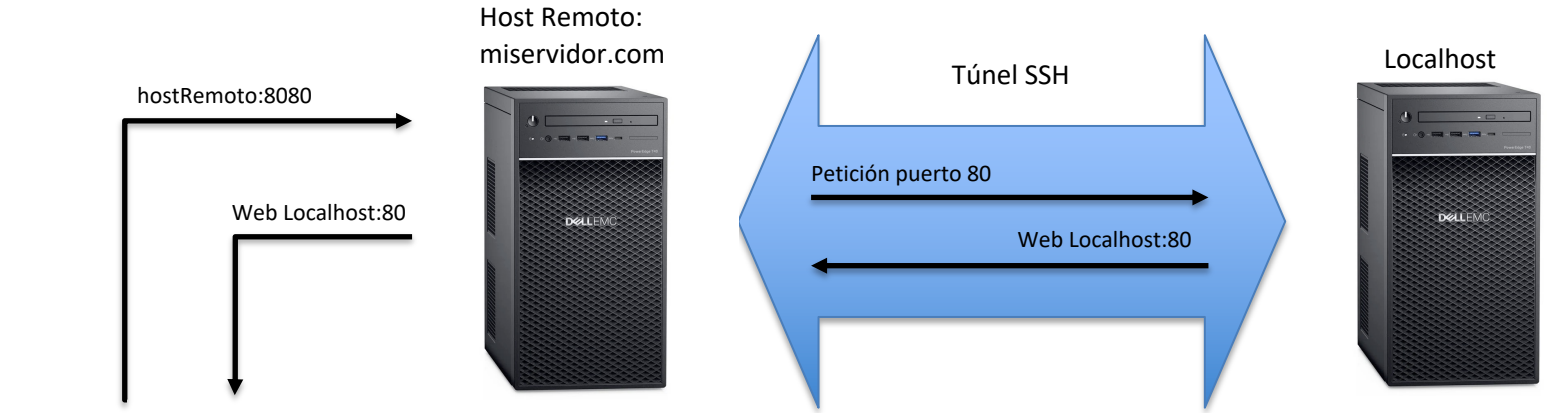
```
$ ssh –R puertoRemoto:localhost:puertoLocal usuario@hostRemoto
```

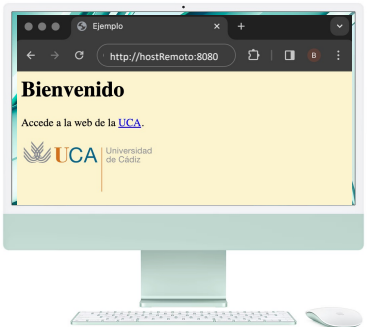
- Ejemplo

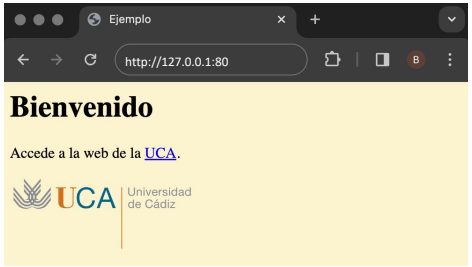
```
$ ssh –R 8080:localhost:80 usuario@hostRemoto.com
```

Tunneling puerto remoto

```
$ ssh –R 8080:localhost:80 usuario@miservidor.com
```







Tunneling

- Túnel para X11
  - Configurar servidor ssh

    ```
    X11Forwarding yes
    ```

  - Conectar al servidor remoto habilitando reenvío X11

    ```
    $ ssh –X hostRemoto
    ```

  - Los programas con interfaz gráfica del servidor se ejecutan como si fuesen locales