1. Administración remota

Estamos habituados a trabajar con aplicaciones con las que interactuamos a través del teclado, ratón y monitor, sentados delante de nuestro equipo. Sin embargo, en el día a día de un administrador de sistemas aparecen situaciones en las que es necesario acceder a equipos remotos, pudiendo estar éstos en nuestra misma red, en una red distinta dentro del mismo edificio, en diferentes edificios de la empresa o incluso en ciudades o países diferentes.

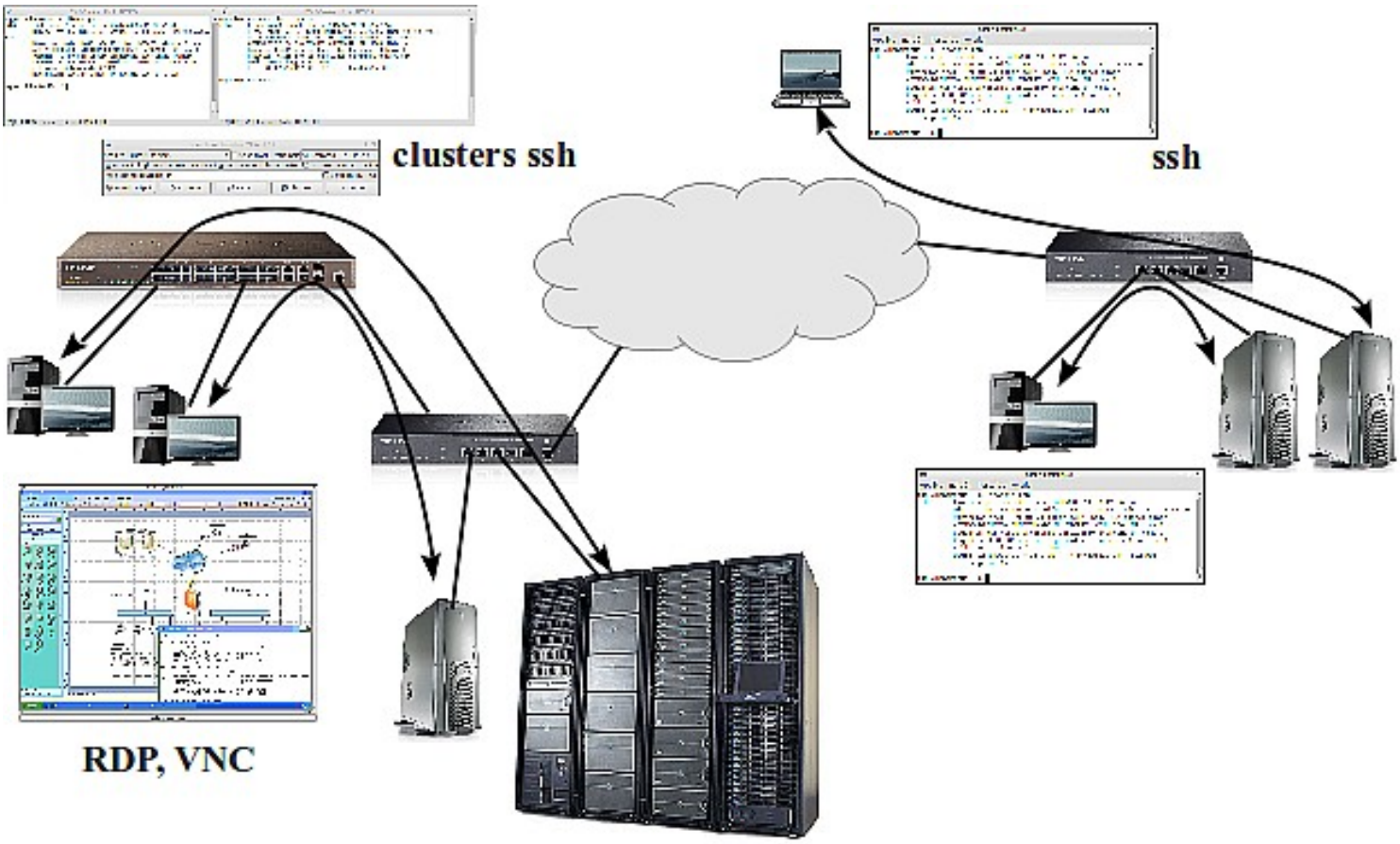

Fig. Ejemplos de sistemas de administración remota

Para poder realizar las tareas necesarias en los equipos remotos evitando costosos desplazamientos se han creado diferentes herramientas de administración remota. Estas herramientas permiten establecer, usando los protocolos TCP/IP, un canal de comunicaciones entre nuestro equipo local y el equipo remoto a través del cual se enviarán órdenes para ejecutarse en la máquina remota y se recibirán resultados para mostrarse en la máquina local.

Estas herramientas de administración remoto siguen el modelo cliente-servidor y se suelen clasificar en base a su forma de trabajo en tres grandes grupos:

- **Modo Consola:**
  - El cliente recibe una consola donde podrá teclear comandos como si estuviese sentado delante del equipo remoto. Los comandos se enviarán al equipo remoto donde se ejecutarán y los resultados serán devueltos para mostrarlos al cliente en su equipo local.
  - Ejemplos: telnet, rlogin y ssh, todos basados en los protocolos del mismo nombre.
- **Modo Interfaz Gráfico:**
  - El cliente accederá al equipo remoto trabajando con una interfaz gráfica que le permitirá trabajar con el mismo escritorio que vería si estuviese sentado delante del equipo remoto.
  - Sus necesidades de ancho de banda son más elevadas que las de modo consola

- Ejemplos: X-Terminal basado en el protocolo XDMP, UltraVNC y tightVNC basados en el protocolo VNC, Terminal Server basado en el protocolo RDP, Teamviewer que usa un protocolo propietario.

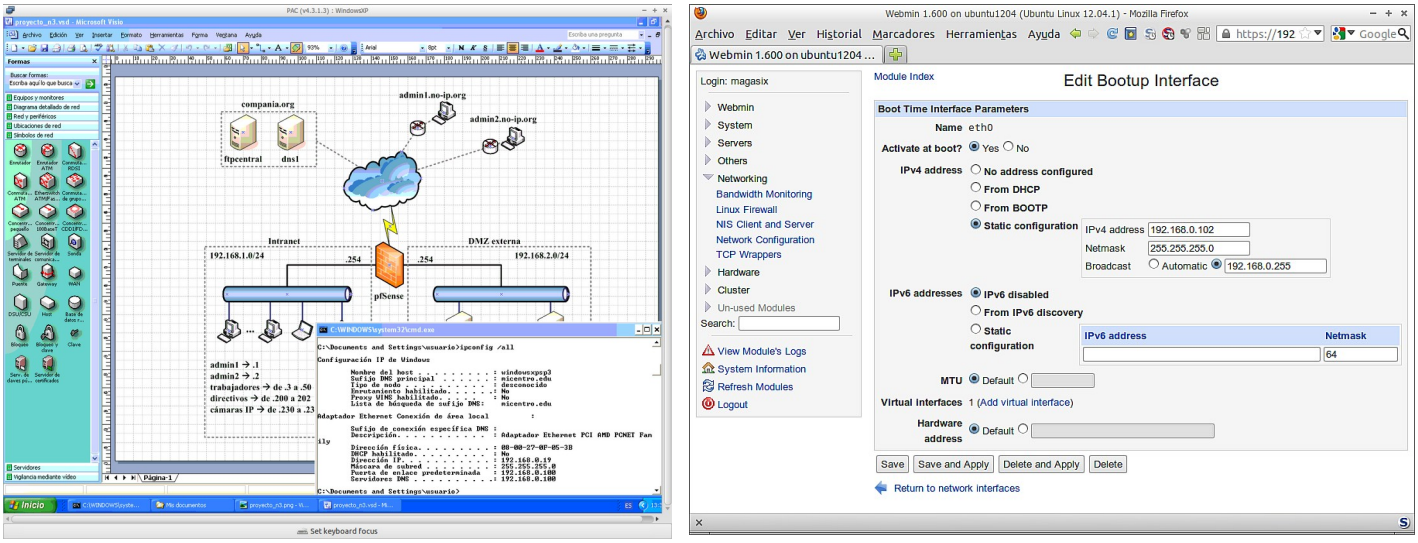

Fig. Ejemplo de acceso por RDP y por Webmin.

- **Modo Web:**
  - El cliente accede al equipo remoto usando un navegador web para controlarlo. Aquí surgen dos posibilidades:
    - Escritorio remoto: el navegador nos muestra el escritorio remoto que podemos controlar usando el teclado y ratón del equipo local. Distintas implementaciones de VNC y RDP permiten trabajar vía navegador, además del modo interfaz gráfico.
    - Interfaz Web: usando el navegador accedemos a una aplicación de gestión basada en web instalada en el equipo remoto. Este método es ampliamente utilizado en dispositivos de red como switches, routers, firewalls, impresoras, …, y en el mundo UNIX/Linux un conocido ejemplo es webmin.

2. Modo Consola: telnet y rlogin

Las herramientas de administración remota en modo consola permiten acceder a una consola y ejecutar comandos en el equipo remoto; por tanto, aunque visualmente no sean atractivas permiten aprovechar toda la potencia de la línea de comandos, además de no consumir tantos recursos de red como las de interfaz gráfico.

- **TELNET (TELecommunication NETwork):**
  - El nombre hace referencia tanto al protocolo como al cliente y el servidor.
  - Se basa en el modelo cliente-servidor:
    - Usa tcp a nivel de transporte.
    - El servidor escucha por defecto en el puerto tcp/23.
  - Antes de obtener la línea de comandos:
    - Se intercambian parámetros, como tipo de terminal usado por el cliente.
    - El cliente debe autenticarse indicando un nombre de usuario/contraseña.
  - Una vez conseguida la línea de comandos el servidor recibe las pulsaciones del cliente, las reenvía y se muestran al cliente (*eco*). Además, el servidor interpreta las pulsaciones (p.e. un comando para ejecutar) y devuelve el resultado al cliente.
  - Existe soporte tanto en Microsoft Windows como en UNIX/Linux.

- Toda la sesión telnet va sin cifrar; es decir, toda la información intercambiada entre el cliente y el servidor va en claro por la red. Por este motivo no se considera seguro y no se recomiendo su uso. En la siguiente imagen puede verse la salida en Wireshark de un flujo tcp asociado a una sesión de telnet (observar la información en claro y el eco):

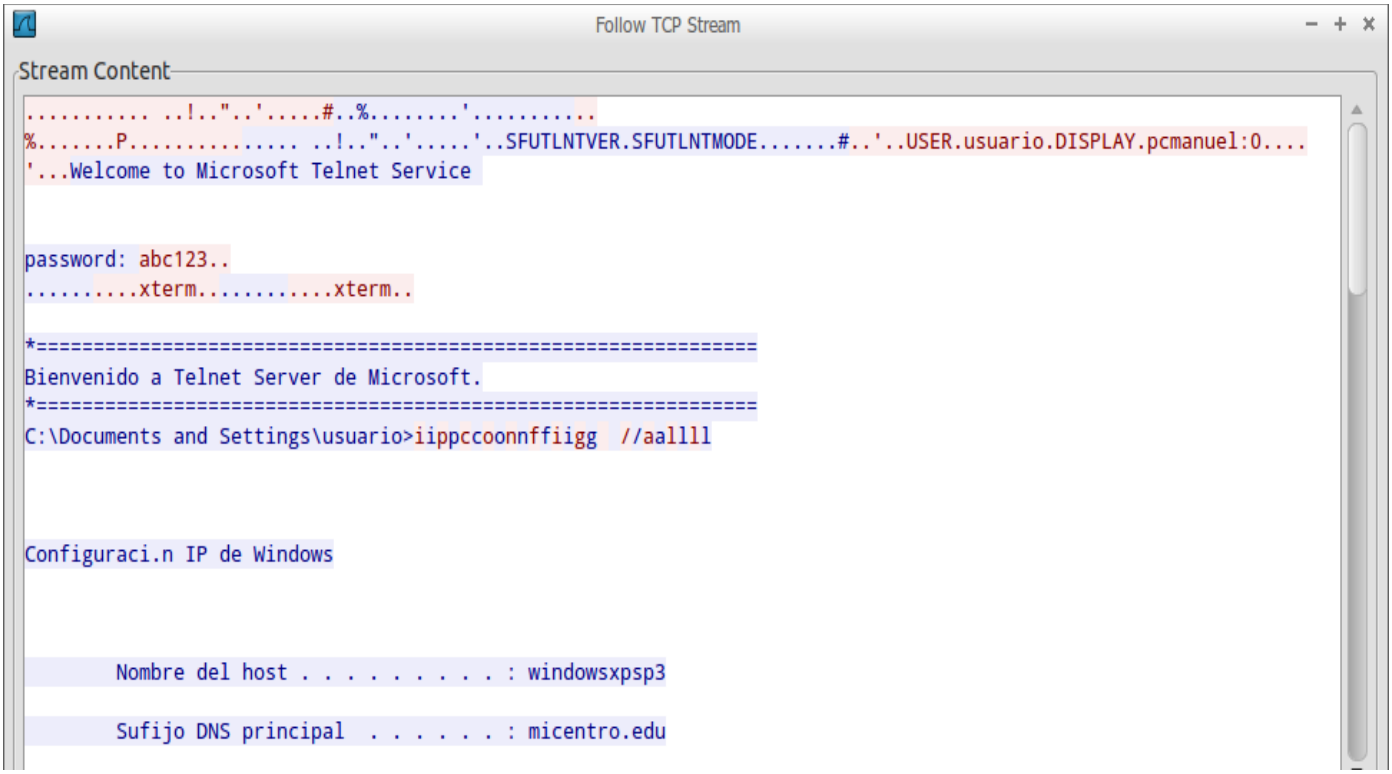

Fig. Captura con Wireshark de una sesión telnet

▪ RLOGIN[1] (Remote LOGIN):

- Funcionamiento similar al de telnet pero sin necesidad de utilizar contraseñas para iniciar la sesión. La autenticación no se basa en contraseña sino en los conceptos de hosts y usuarios confiables:
  - Los hosts desde los que se permite iniciar la sesión se definen en el archivo /etc/hosts.equiv. Es posible indicar además, que desde un equipo se permite el acceso a unos determinados usuarios.
  - Cada usuario puede definir en su archivo $HOME/.rhosts desde que equipos (y con que cuenta) se permite iniciar la sesión. Es el equivalente de usuario del archivo /etc/hosts.equiv
- Se basa en el modelo cliente-servidor:
  - Usa tcp a nivel de transporte.
  - El servidor escucha por defecto en el puerto tcp/513.
- Soporte tanto en Microsoft Windows como en UNIX/Linux.
- Seguridad:
  - Toda la sesión rlogin va sin cifrar por lo que no se considera seguro.
  - Las relaciones de confianza basadas en ficheros como el .rhosts no son seguras (p.e. suplantación de un equipo en la red); por lo que de usarse, únicamente se debería hacer en entornos muy controlados.

3. Criptografía

La criptografía es la práctica y el estudio de la **ocultación de información**. Es la ciencia de cifrar y descifrar cierta información mediante técnicas especiales y se emplea frecuentemente para

[1] Comandos-R: rsh y rcp → precursores de ssh y scp

permitir un intercambio de mensajes que sólo puedan ser leídos por personas a las que van dirigidos y que poseen los medios para descifrarlos.

La criptografía que actualmente se utiliza data de los años 70, aunque ya se utilizaban técnicas criptográficas en la antigüedad. Estas técnicas se aplican cada día en el mundo de la tecnología en multitud de usos como las comunicaciones militares, protocolo https, protocolo ssh, envío de correos electrónicos cifrados o las telecomunicaciones en general.

La información original que debe protegerse se denomina **texto plano** (**texto en claro)**. El cifrado es el proceso de convertir el texto plano en un texto imposible de leer llamado **texto cifrado o criptograma**. Para obtener un texto cifrado, se aplica un algoritmo de cifrado, utilizando una clave, al texto plano.

Fig. Cifrado de un texto

Para recuperar el mensaje original, se tomará el criptograma (texto cifrado), la clave y se aplicará el algoritmo de descifrado:

Fig. Descifrado de un criptograma

Entre los principales objetivos que busca la criptografía están:

- **Confidencialidad:** la información se revela únicamente a los usuarios autorizados. Por ejemplo, al usar https al acceder a un sistema de webmail, los datos de usuario/contraseña irán cifrados por un túnel de comunicaciones establecido entre el navegador del usuario y el servidor. Se logra que el navegador y el servidor puedan intercambiar información pero que esa información no sea legible para los demás (si un sniffer capturase los paquetes de la comunicación se verían los datos cifrados).
- **Integridad:** garantiza la exactitud de la información contra su alteración, pérdida o destrucción. Por ejemplo, cuando se firma electrónicamente un documento, se aporta un 'resumen' que garantiza que si el documento original fuese modificado, se podría detectar la manipulación al no coincidir el resumen del documento original con el resumen del documento modificado.
- **Autenticación:** comprobación de la identidad. Por ejemplo, al usar ssh para conectarse a un equipo remoto, el cliente comprueba la identidad digital del servidor ssh para asegurarse que se está conectando al servidor de verdad y no con un impostor.
- **No repudio (irrenunciabilidad o vinculación):** vincula un documento o transacción a una persona o equipo. Por ejemplo, en la firma electrónica de documentos se garantiza el no repudio en el emisor; es decir, se emplean mecanismos que garantizan que un documento firmado electrónicamente por una persona usando su identidad digital está asociado a esa persona, no siendo posible que fuese firmado por otra.

Dependiendo del número de claves utilizadas por los algoritmos de cifrado/descifrado, existen dos tipos de métodos criptográficos:

- **Criptografía de clave simétrica**, que utiliza una única clave para cifrar/descifrar.
- **Criptografía de clave asimétrica**, que utiliza dos claves.

Como se verá más adelante, existen sistemas llamados de **criptografía híbrida**, que emplean los dos sistemas (simétrico y asimétrico) procurando escoger lo mejor de cada uno (seguridad y rendimiento).

3.1 Criptografía de clave simétrica

Se usa una única clave para cifrar y descifrar mensajes. Una vez que las dos partes que se van a comunicar poseen la misma clave, el remitente cifra un mensaje con ella, lo envía al destinatario y éste lo descifra con la misma clave.

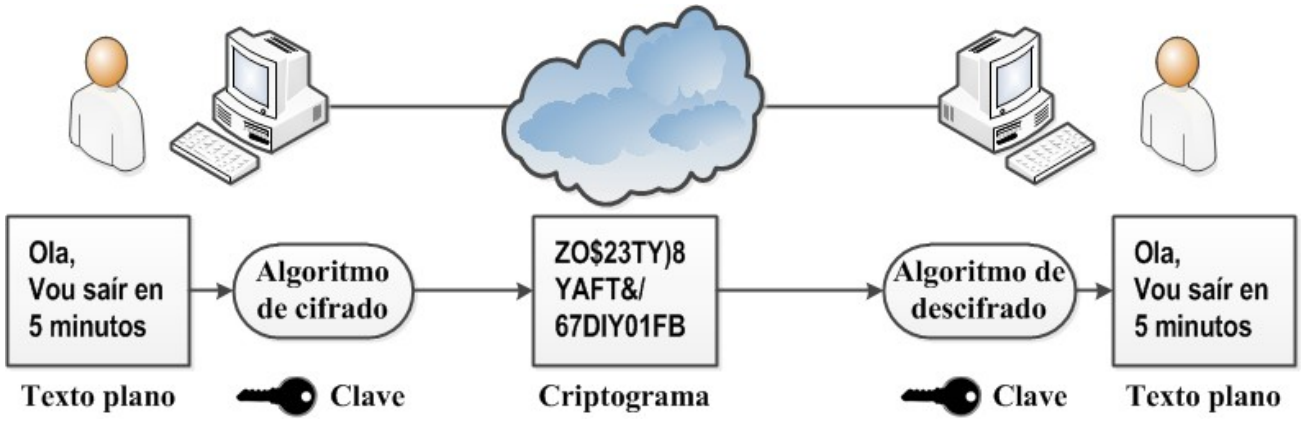

Fig. Cifrado de información mediante criptografía de clave simétrica

Un problema que tienen todos los sistemas de clave simétrica es el intercambio de la clave; ya que antes de poder enviar información, los participantes tienen que intercambiarse la clave. Si la clave cae en malas manos, toda comunicación que use esa clave comprometida no será segura porque personas no autorizadas podrán acceder a los contenidos transmitidos. Si se encuentra un sistema seguro para intercambiar la clave, este sistema es el más eficiente.

Ejemplos de algoritmos de cifrado simétricos son DES, AES, 3DES, Blowfish e IDEA. A modo de ejemplo, el algoritmo AES (Advanced Encryption Standard) permite usar claves de 128, 192 e 256 bits (cuanto más grande sea la clave, más seguridad se tiene). AES es un estándar de cifrado del Gobierno de los Estados Unidos de América y para los documentos clasificados como TOP SECRET es obligatorio el uso de claves de 192 o 256 bits.

3.2 Criptografía de clave asimétrica

Los sistemas criptográficos de clave simétrica tienen el problema de la distribución de la clave entre los participantes de la comunicación. En los sistemas de clave asimétrica no existe este problema y cada participante tiene una pareja de claves propias:

- **Clave privada:**
  - Su propietario debe mantenerla en secreto a toda costa.
  - Permite cifrar mensajes (que podrán ser descifrados con la clave pública).
  - Permite descifrar mensajes cifrados con la clave pública del propietario.
- **Clave pública:**
  - Debe repartirse a todos los usuarios con los que se quiera comunicar.
  - Se usa para enviar mensajes cifrados al propietario, que sólo él podrá descifrar usando su clave privada.
  - Se usa para descifrar mensajes enviadas por el propietario y cifrados con la clave privada.

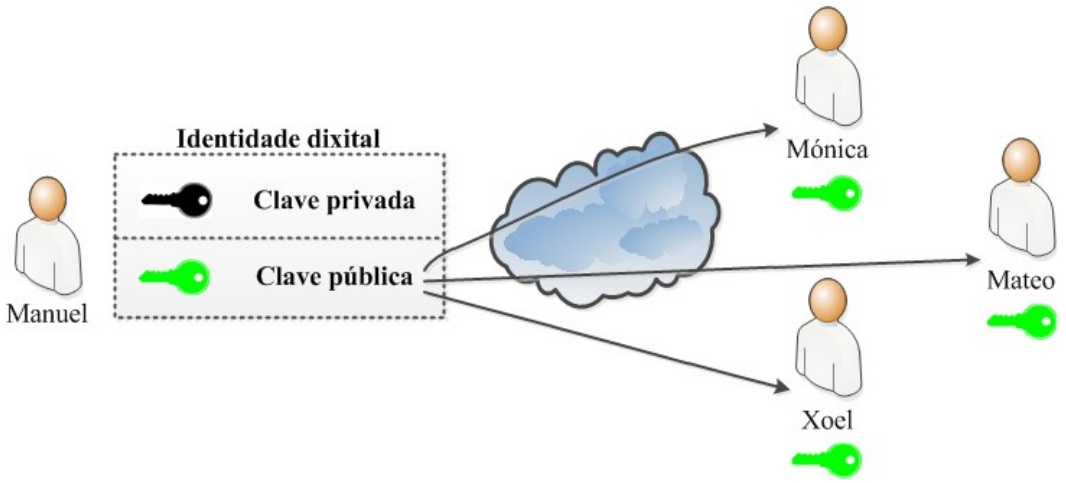

Fig. Reparto de la clave pública en criptografía de clave asimétrica

En la siguiente imagen se puede ver lo que sucede si Mónica envía un mensaje a Manuel cifrándolo con la clave pública de Manuel. Lo que se consigue, es que únicamente Manuel pueda descifrarlo usando su clave privada (la de Manuel). Nadie, salvo el que tenga la clave privada de Manuel, podrá descifrar el mensaje; por lo tanto, se consiguió la confidencialidad.

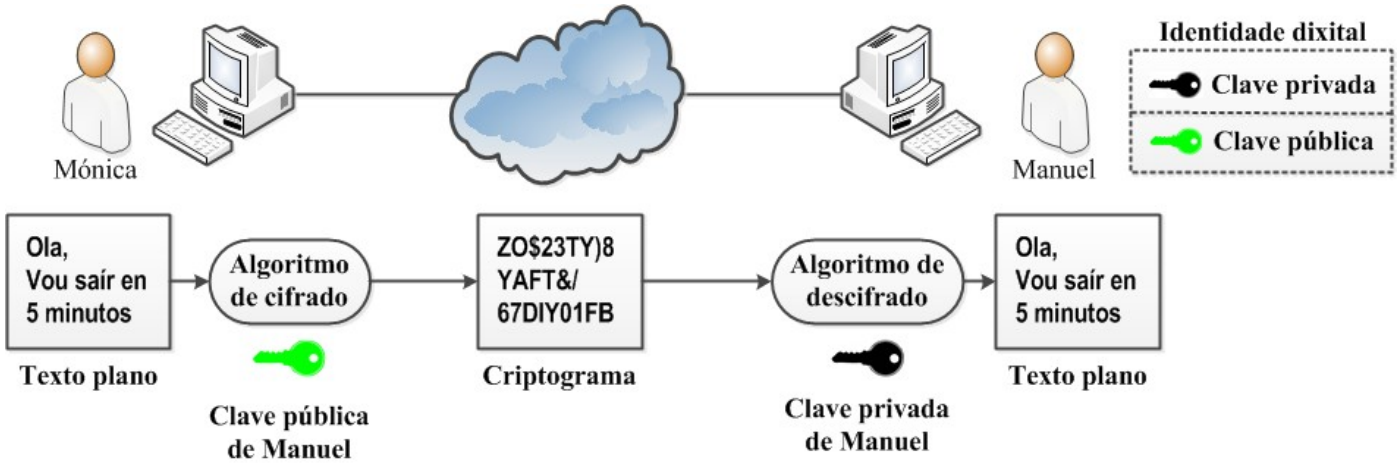

Fig. Confidencialidad de la información mediante criptografía de clave asimétrica

Ahora Manuel envía un mensaje cifrado con su clave privada a Mónica. Mónica, usando la clave pública de Manuel, será capaz de descifrar el mensaje y tendrá la seguridad de que el mensaje fue enviado por Manuel; ya que nadie salvo Manuel, tiene la clave privada (de Manuel). Se logra la identificación y la vinculación (no repudio) del remitente.

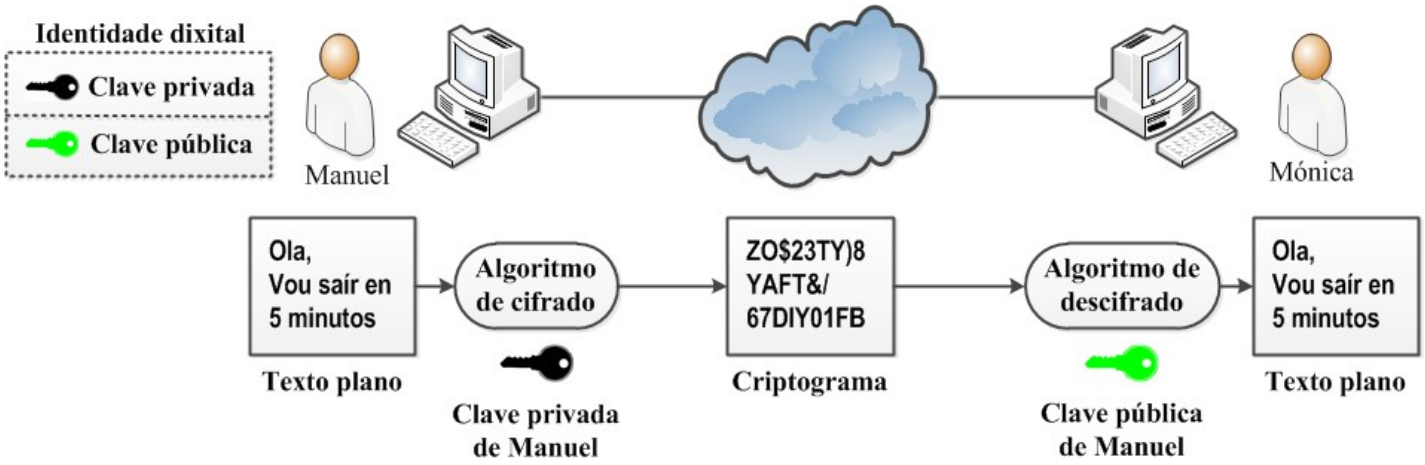

Fig. Identificación y vinculación del remitente mediante criptografía de clave asimétrica

Una vez entendidas las posibilidades que ofrece el uso de las claves privadas y públicas de Manuel, se plantea la situación donde Manuel y Mónica quieren comunicarse buscando a la vez confidencialidad, identificación y vinculación. Para lograrlo hay que cumplir con las siguientes condiciones:

- Manuel tiene su clave privada, que no comparte con nadie.
- Manuel envía su clave pública a Mónica.
- Mónica tiene su clave privada, que no comparte con nadie.
- Mónica envía su clave pública a Manuel.
- Mensajes de Manuel a Mónica:
  - Manuel cifra el mensaje con su clave privada y después lo cifra con la clave pública de Mónica.
  - El mensaje doblemente cifrado se envía a Mónica.
  - Una vez que le llega el mensaje a Mónica, ésta lo descifra con su clave privada y el resultado procede a descifrarlo con la clave pública de Manuel. Una vez aplicado este procedimiento, Mónica recupera el mensaje original de Manuel.
- Mensajes de Mónica a Manuel:
  - Mónica cifra el mensaje con su clave privada y después lo cifra con la clave pública de Manuel.
  - El mensaje doblemente cifrado se envía a Manuel.
  - Una vez que le llega el mensaje a Manuel, éste lo descifra con su clave privada y el resultado procede a descifrarlo con la clave pública de Mónica. Una vez aplicado este procedimiento, Manuel recupera el mensaje original de Mónica.

Lo que se consigue con este sistema es:

- Al cifrar en primer lugar con la clave privada del remitente, el destinatario podrá estar seguro que el mensaje realmente procede del remitente; y por lo tanto, se garantiza la identificación y vinculación del remitente.
- Al cifrar en segundo lugar con la clave pública del destinatario, se asegura que únicamente el destinatario puede ver el contenido del mensaje (gracias a que la única clave que puede descifrar el mensaje es la privada del destinatario); y por lo tanto, se garantiza la confidencialidad del envío.

El proceso a seguir en el envío de mensajes de Manuel a Mónica puede verse ejemplificado en la siguiente imagen.

Los métodos criptográficos garantizan:

- Que esa pareja de claves sólo se puede generar una vez, de modo que se puede asumir que no es posible que dos personas obtuviesen casualmente la misma pareja de claves.
- Que a partir de la clave pública no se puede deducir nada de la clave privada.
- Que lo que cifra la clave pública sólo puede ser descifrado con la privada y lo que cifra la clave privada sólo lo descifra la pública.

Ahora el intercambio de claves ya no es problema; ya que, la pública está pensada para repartirse. Lo que si es crítico es mantener la clave privada a buen recaudo. Ejemplos de métodos de cifrado asimétricos son Diffie-Hellman, RSA, DSA y ElGamal.

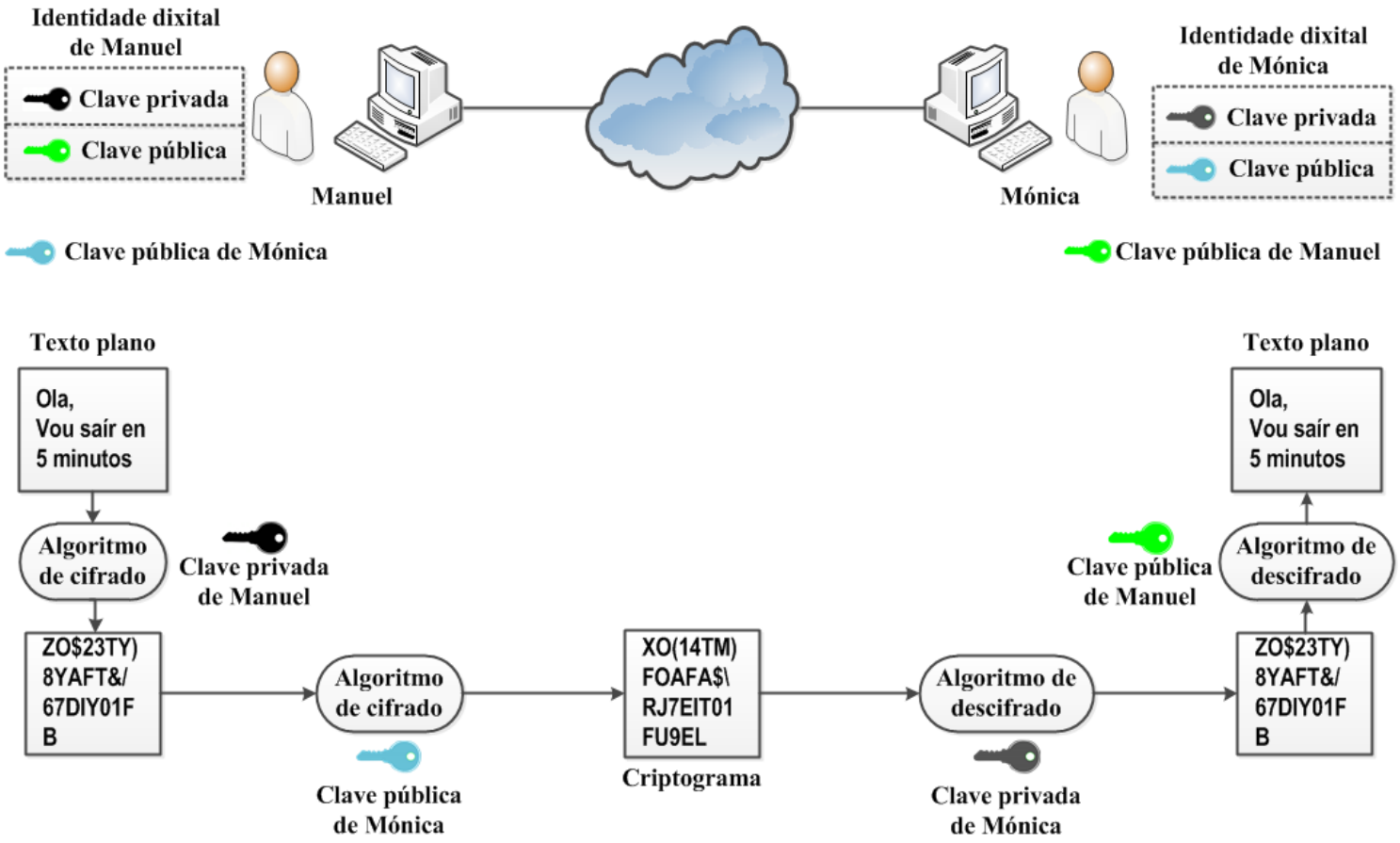

Fig. Confidencialidad de la información, identificación y vinculación del remitente usando criptografía asimétrica

3.3 Criptografía híbrida

Debido a que los sistemas de cifrado asimétricos son computacionalmente más costosos que los simétricos y que estos últimos tienen el problema del intercambio de claves, se inventaron los sistemas de criptografía híbrida. En estos sistemas híbridos, al comienzo de la comunicación se usa un sistema de claves asimétricas para intercambiar de forma segura una clave simétrica de sesión, que se usará para cifrar el resto de la comunicación de forma más eficiente. En función de la implementación, la clave simétrica puede usarse en toda la sesión o a lo largo de la misma cambiar y usar varias.

Un ejemplo de implementación práctica de este tipo de sistema se puede encontrar en los protocolos SSH y https.

En la siguiente figura puede verse un resumen del proceso:

- Paso 1: Manuel envía la clave pública a Mónica.
- Paso 2: Mónica genera una clave de sesión que es cifrada usando la clave pública de Manuel y la envía a Manuel. Este procede a descifrarla usando su clave privada.
- Paso 3: una vez que los dos participantes de la comunicación ya conocen la clave de sesión, pueden proceder a enviar mensajes cifrados y descifrarlos usando la misma clave (criptografía simétrica).

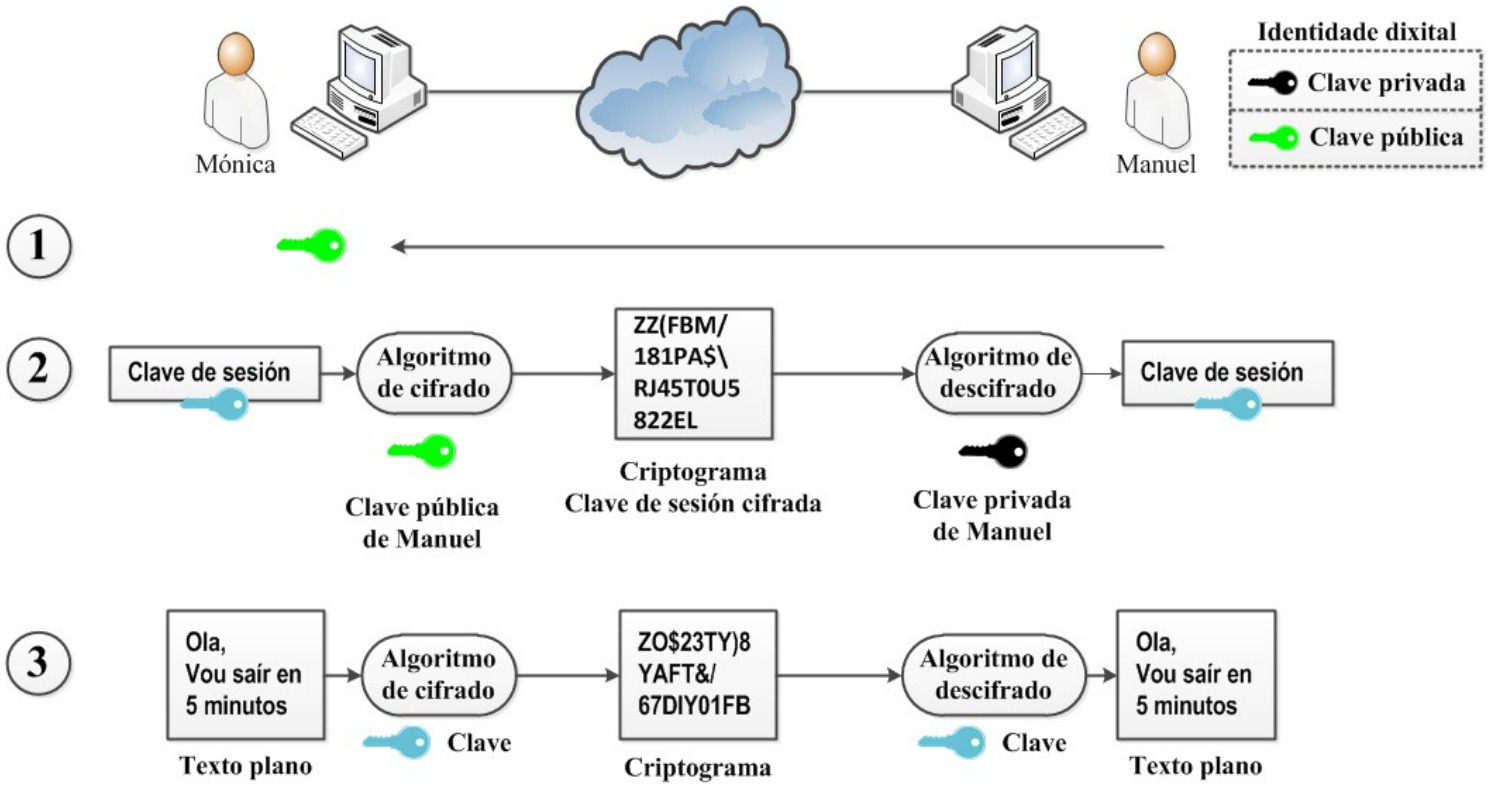

Fig. Funcionamiento de un sistema de criptografía híbrida

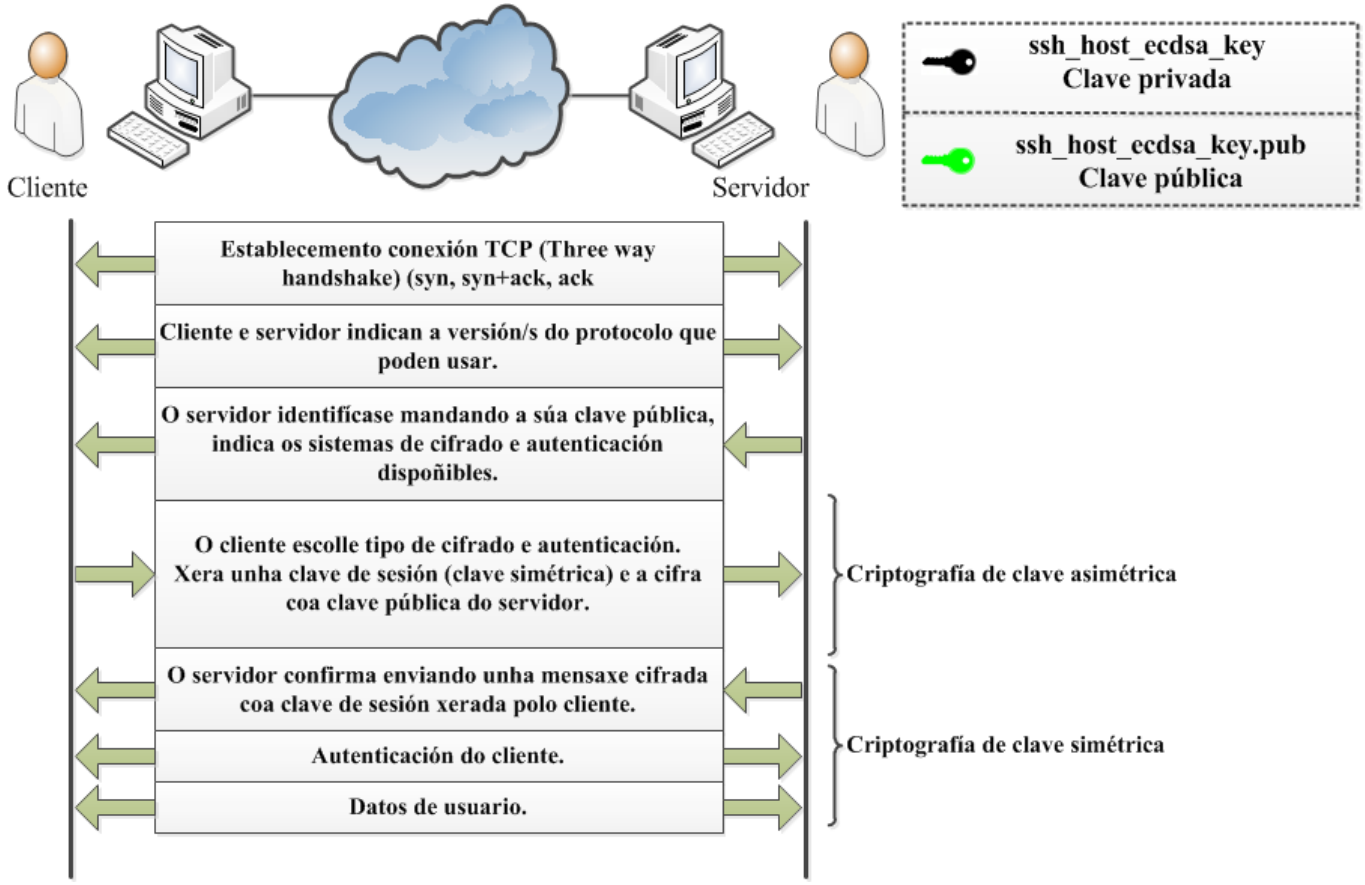

Fig. Establecimiento de una conexión SSH

4. Modo Consola: SSH (Secure Shell)

El nombre hace referencia al protocolo de nivel de aplicación que permite conectarse de forma remota a un equipo obteniendo una línea de comandos para su administración. Se trata de un protocolo basado en el modelo cliente-servidor, donde el servidor escucha por defecto en el puerto tcp/22, habiendo soporte tanto en Microsoft Windows como en UNIX/Linux.

SSH emplea la criptografía para encargarse (de forma transparente para el usuario) de las operaciones necesarias para conseguir:

- **Privacidad:** la información que circula por la red está protegida al usarse técnicas de **cifrado** fuerte. Los datos son 'transformados' de forma que son entendibles únicamente por su legítimo destinatario; por tanto, aunque un sniffer los capturase no se podrían recuperar. En la siguiente imagen se puede ver un paquete SSH; donde al ir cifrado su contenido, no es legible.

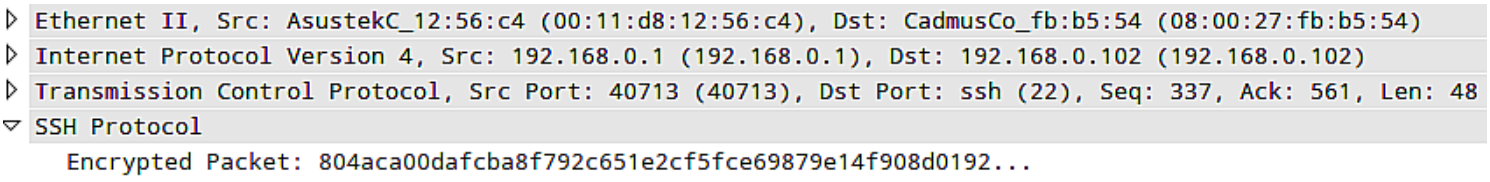

Fig. Captura en Wireshark de un paquete SSH

- **Integridad:** se garantiza que los datos no se han alterado durante su viaje por la red. Si un tercero modifica los datos en tránsito el destinatario será consciente de ello. SSH se encarga de verificar que los datos no se han modificado y que además proceden realmente del otro extremo de la conexión.
- **Autenticación:**[2] antes de poder iniciar una sesión en el equipo remoto se comprueban las identidades del servidor y del cliente:
  - **Server Authentication:** El cliente comprueba que la identidad del servidor ssh para asegurarse que se está conectando al servidor de verdad y no con un impostor.

```
manuel@lubuntu:~$ ssh magasix@192.168.1.254
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
a9:c4:8c:cb:b8:0c:d3:10:e8:83:1b:4b:02:5d:3b:65.
Please contact your system administrator.
Add correct host key in /home/manuel/.ssh/known_hosts to get rid of this
message.
Offending ECDSA key in /home/manuel/.ssh/known_hosts:1
  remove with: ssh-keygen -f "/home/manuel/.ssh/known_hosts" -R 192.168.1.254
ECDSA host key for 192.168.1.254 has changed and you have requested strict
checking.
Host key verification failed.
manuel@lubuntu:~$
```

Fig. Error de autenticación por no coincidir las claves públicas

  - **User Authentication:** el servidor comprueba la identidad del usuario que quiere conectarse. SSH permite diferentes mecanismos de autenticación de usuarios; negociando el cliente y el servidor, cuál se va a usar. Destacan los basados en:
    - **Usuario-contraseña:** El usuario envía una contraseña que es verificada por el servidor. A diferencia de telnet, en ssh tanto el envío del usuario/contraseña como toda la sesión está cifrada
    - **Claves (keys):** El servidor comprueba la identidad de usuario gracias a unas claves o identidades digitales.

```
manuel@lubuntu:~$ ls -lahF .ssh
```

[2] Identificación: procedimiento por el que un usuario es reconocido dentro del sistema (p.e. a través del nombre del usuario).
Autenticación: comprobación de la identidad en el sistema (p.e. a través de la contraseña de acceso asociada al nombre de usuario).

```
total 48K
drwx------  2 manuel manuel 4,0K feb 14 23:51 ./
drwxr-xr-x 77 manuel manuel  12K feb 20 17:12 ../
-rw-------  1 manuel manuel 3,3K dic  5 13:17 id_rsa
-rw-------  1 manuel manuel 3,2K dic  5 13:58 id_rsa_asir01
-rw-r--r--  1 manuel manuel  756 dic  5 13:58 id_rsa_asir01.pub
-rw-------  1 manuel manuel 3,3K dic  4 15:44 id_rsa_magasix
-rw-r--r--  1 manuel manuel  743 dic  4 15:44 id_rsa_magasix.pub
-rw-r--r--  1 manuel manuel  759 dic  5 13:17 id_rsa.pub
-rw-------  1 manuel manuel 2,2K feb 19 02:30 known_hosts
-rw-------  1 manuel manuel 2,0K feb 10 03:38 known_hosts.old
manuel@lubuntu:~$ ssh -i .ssh/id_rsa magasix@192.168.56.253
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.13.0-76-generic x86_64)
 * Documentation:  https://help.ubuntu.com/
  System information as of Sat Feb 20 18:48:14 CET 2016
  System load:  0.09              Processes:           72
  Usage of /:   7.7% of 19.07GB   Users logged in:     1
  Memory usage: 11%               IP address for eth0: 192.168.56.253
  Swap usage:   0%
  Graph this data and manage this system at:
    https://landscape.canonical.com/
Last login: Sat Feb 20 18:48:14 2016 from 192.168.56.1
magasix@server:~$
```

Fig. Ejemplo de autenticación del usuario magasix en un equipo remoto usando su identidad digital (pareja de claves privada/pública).

- **Control de Acceso:** se pueden restringir los accesos aplicando controles por usuario, máquina y usuario-máquina. En el siguiente ejemplo, únicamente se permite el acceso por ssh al usuario magasix desde la máquina 192.168.56.100. De no cumplirse ambas condiciones a la vez, no se puede acceder por ssh como puede verse en el archivo de log:

```
/etc/ssh/sshd_config
...
AllowUsers magasix@192.168.56.100
...

manuel@lubuntu:~$ ssh magasix@192.168.56.253
magasix@192.168.56.253's password:
Permission denied, please try again.
magasix@192.168.56.253's password:
Permission denied, please try again.
magasix@192.168.56.253's password:
Received disconnect from 192.168.56.253: 2: Too many authentication failures for
magasix
manuel@lubuntu:~$

Feb 20 19:24:31 server sshd[2442]: User magasix from 192.168.56.1 not allowed
because not listed in AllowUsers
Feb 20 19:24:31 server sshd[2442]: input_userauth_request: invalid user magasix
[preauth]
Feb 20 19:24:33 server sshd[2442]: pam_unix(sshd:auth): authentication failure;
logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.56.1  user=magasix
```

Fig. Ejemplo de directiva de control de acceso en ssh

- **Ejecución Remota y Segura de Comandos:** Aunque el uso más habitual de ssh es el de obtener una línea de comandos en el equipo remoto, también ofrece la posibilidad de ejecutar un único comando; por ejemplo:

```
manuel@lubuntu:~$ ssh magasix@192.168.56.253 ethtool eth0
Settings for eth0:
    Supported ports: [ TP ]
```

```
    Supported link modes:   10baseT/Half 10baseT/Full
                            100baseT/Half 100baseT/Full
                            1000baseT/Full
    Supported pause frame use: No
    Supports auto-negotiation: Yes
    Advertised link modes:  10baseT/Half 10baseT/Full
                            100baseT/Half 100baseT/Full
                            1000baseT/Full
    Advertised pause frame use: No
    Advertised auto-negotiation: Yes
    Speed: 1000Mb/s
    Duplex: Full
    Port: Twisted Pair
    PHYAD: 0
    Transceiver: internal
    Auto-negotiation: on
    MDI-X: off (auto)
    Current message level: 0x00000007 (7)
                   drv probe link
    Link detected: yes
Cannot get wake-on-lan settings: Operation not permitted
manuel@lubuntu:~$
```

Fig. Ejecución remota de comandos en el equipo 192.168.56.253 por ssh

- **Transferencia de Archivos Segura:** Permite copiar ficheros entre dos equipos de forma segura (autenticación y transferencia cifradas) con scp. Además con ssh y sftp se puede tener un reemplazo al servicio FTP donde todo el tráfico va en claro.

```
manuel@lubuntu:~$ scp webmin_1.780_all.deb magasix@192.168.56.253:
webmin_1.780_all.deb                                  100%   27MB  26.7MB/s   00:00
manuel@lubuntu:~$ scp magasix@192.168.56.253:/var/log/iptables.log
iptables_20_02_16.log
iptables.log                                          100%  468     0.5KB/s   00:00
manuel@lubuntu:~/Descargas$
```

Fig. Ejemplo de copia de archivos de local a remoto y de remoto a local usando scp

- **Tunneling o Port Forwarding:** SSH permite crear conexiones cifradas entre dos equipos por donde circula tráfico procedente de otras aplicaciones. Esto permitiría usar de forma segura y sin hacer modificaciones, protocolos no seguros por que en su diseño original no se contempló el cifrado.

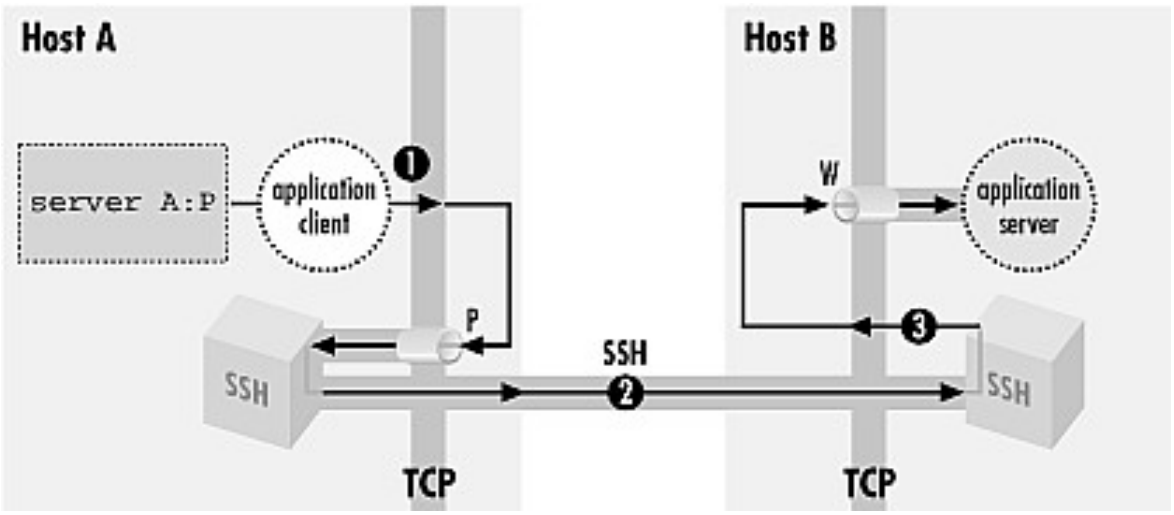

Fig. Esquema de funcionamiento de un túnel SSH