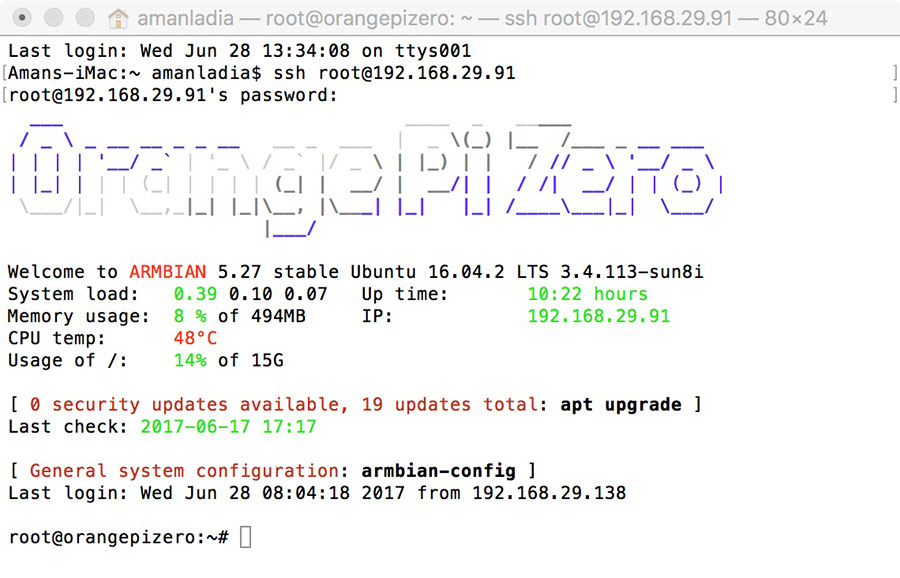
**¿Cómo funciona el SSH?**



SSH o Secure Shell, es un protocolo de administración remota que le permite a los usuarios controlar y modificar sus servidores remotos a través de Internet a través de un mecanismo de autenticación.

Proporciona un mecanismo para autenticar un usuario remoto, transferir entradas desde el cliente al host y retransmitir la salida de vuelta al cliente. El servicio se creó como un reemplazo seguro para el Telnet sin cifrar y utiliza técnicas criptográficas para garantizar que todas las comunicaciones hacia y desde el servidor remoto sucedan de manera encriptada.

La imagen de abajo muestra una ventana típica de SSH. Cualquier usuario de Linux o macOS puede usar SSH en su servidor remoto directamente desde la ventana del terminal. Los usuarios de Windows pueden aprovechar los [**clientes SSH como Putty.**](https://www.hostinger.es/tutoriales/crear-tunel-ssh-putty/) Puedes ejecutar comandos shell de la misma manera que lo harías si estuvieras operando físicamente el equipo remoto.



Este tutorial de SSH cubrirá los conceptos básicos de cómo funciona, junto con las tecnologías subyacentes utilizadas por el protocolo SSH para ofrecer un método seguro de acceso remoto. Cubrirá las diferentes capas y tipos de cifrado utilizados, junto con el propósito de cada capa.

# Tabla de Contenidos

* [¿Cómo funciona SSH?](https://www.hostinger.es/tutoriales/que-es-ssh#%C2%BFComo_funciona_SSH)
* [Comprendiendo las diferentes técnicas de cifrado](https://www.hostinger.es/tutoriales/que-es-ssh#Comprendiendo_las_diferentes_tecnicas_de_cifrado)
  + [Cifrado Simétrico](https://www.hostinger.es/tutoriales/que-es-ssh#Cifrado_Simetrico)
  + [Cifrado Asimétrico](https://www.hostinger.es/tutoriales/que-es-ssh#Cifrado_Asimetrico)
  + [Hashing](https://www.hostinger.es/tutoriales/que-es-ssh#Hashing)
* [¿Cómo funciona el protocolo SSH con estas técnicas de cifrado?](https://www.hostinger.es/tutoriales/que-es-ssh#%C2%BFComo_funciona_el_protocolo_SSH_con_estas_tecnicas_de_cifrado)
  + [Negociación de cifrado de sesión](https://www.hostinger.es/tutoriales/que-es-ssh#Negociacion_de_cifrado_de_sesion)
  + [Autenticación del usuario](https://www.hostinger.es/tutoriales/que-es-ssh#Autenticacion_del_usuario)

**¿Cómo funciona SSH?**

Si usas Linux o Mac, entonces usar el protocolo SSH es muy fácil. Si utilizas Windows, deberás utilizar un cliente SSH para abrir conexiones SSH. El cliente SSH más popular es PuTTY, puedes [**aprender más acerca de él aquí**](https://www.hostinger.es/tutoriales/crear-tunel-ssh-putty/).

Para usuarios de Mac y Linux, dirígete a tu programa de **terminal** y sigue este procedimiento:

El comando SSH consta de 3 partes distintas:

ssh {user}@{host}

El comando de clave SSH le indica a tu sistema que desea abrir una Conexión de Shell Segura y cifrada. **{user}** representa la cuenta a la que deseas acceder. Por ejemplo, puede que quieras acceder al usuario **root**, que es básicamente para el administrador del sistema con derechos completos para modificar cualquier cosa en el sistema. **{host}** hace referencia al equipo al que quieres acceder. Esto puede ser una dirección IP **(por ejemplo, 244.235.23.19)** o un nombre de dominio **(por ejemplo, www.xyzdomain.com)**.

Al pulsar enter, se te pedirá que escribas la contraseña de la cuenta solicitada. Al escribirla, nada aparecerá en la pantalla, pero tu contraseña, de hecho, se está transmitiendo. Una vez que hayas terminado de escribir, pulsa enter una vez más. Si tu contraseña es correcta, verás una ventana de terminal remota.

Si quieres aprender más sobre comandos SSH, [**puedes verlos aquí**](https://www.hostinger.com/tutorials/ssh/basic-ssh-commands).

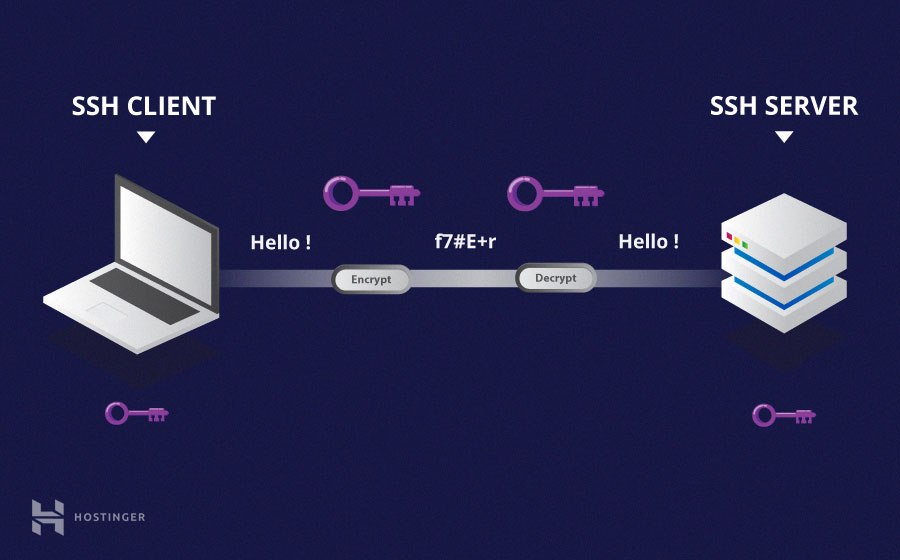
**Comprendiendo las diferentes técnicas de cifrado**

La ventaja significativa ofrecida por el protocolo SSH sobre sus predecesores es el uso del cifrado para asegurar la transferencia segura de información entre el host y el cliente. **Host** se refiere al servidor remoto al que estás intentando acceder, mientras que el **cliente** es el equipo que estás utilizando para acceder al host. Hay tres tecnologías de cifrado diferentes utilizadas por SSH:

1. Cifrado simétrico  
2. Cifrado asimétrico  
3. Hashing

**Cifrado Simétrico**

El cifrado simétrico es una forma de cifrado en la que se utiliza una **clave secreta** tanto para el cifrado como para el descifrado de un mensaje, tanto por el cliente como por el host. Efectivamente, cualquiera que tenga la clave puede descifrar el mensaje que se transfiere.



El cifrado simétrico a menudo se llama **clave compartida** (shared key) o **cifrado secreto compartido**. Normalmente sólo hay una clave que se utiliza, o a veces un par de claves donde una clave se puede calcular fácilmente con la otra clave.

Las claves simétricas se utilizan para cifrar toda la comunicación durante una sesión SSH. Tanto el cliente como el servidor derivan la clave secreta utilizando un método acordado, y la clave resultante nunca se revela a terceros. El proceso de creación de una clave simétrica se lleva a cabo mediante un **algoritmo de intercambio de claves**.

Lo que hace que este algoritmo sea particularmente seguro es el hecho de que la clave nunca se transmite entre el cliente y el host. En lugar de eso, los dos equipos comparten datos públicos  y luego los manipulan para calcular de forma independiente la clave secreta. Incluso si otra máquina captura los datos públicamente compartidos, no será capaz de calcular la clave porque el algoritmo de intercambio de clave no se conoce.

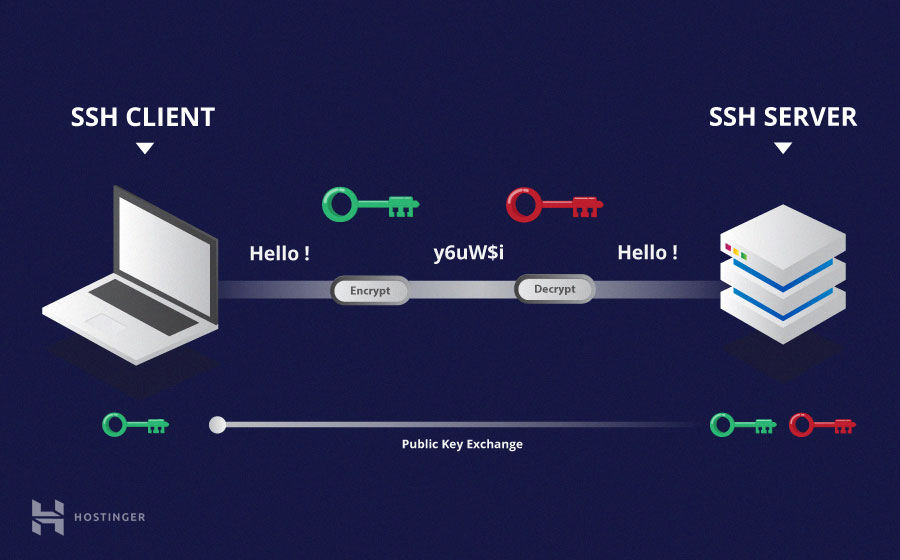
Debe tenerse en cuenta, sin embargo, que el token secreto es específico para cada sesión SSH, y se genera antes de la autenticación del cliente. Una vez generada la clave, todos los paquetes que se mueven entre las dos máquinas deben ser cifrados por la clave privada. Esto incluye la contraseña escrita en la consola por el usuario, por lo que las credenciales siempre están protegidas de los fisgones de paquetes de red.

Existen varios códigos cifrados simétricos, incluyendo, pero no limitado a, AES (Advanced Encryption Standard), CAST128, Blowfish, etc. Antes de establecer una conexión segura, el cliente y un host deciden qué cifrado usar, publicando una lista de cifrados soportados por orden de preferencia. El cifrado preferido de entre los soportados por los clientes que está presente en la lista del host se utiliza como el cifrado bidireccional.

Por ejemplo, si dos máquinas Ubuntu 14.04 LTS se comunican entre sí a través de SSH, utilizarán **aes128-ctr** como su cifrado predeterminado.

**Cifrado Asimétrico**

A diferencia del cifrado simétrico, el cifrado asimétrico utiliza dos claves separadas para el cifrado y el descifrado. Estas dos claves se conocen como la **clave pública** (public key) y la **clave privada** (private key). Juntas, estas claves forman el par de **claves pública-privada** (public-private key pair).



La clave pública, como sugiere el nombre, se distribuye abiertamente y se comparte con todas las partes. Si bien está estrechamente vinculado con la clave privada en términos de funcionalidad, la clave privada no se puede calcular matemáticamente desde la clave pública. La relación entre las dos claves es altamente compleja: un mensaje cifrado por la clave pública de una máquina, sólo puede ser descifrado por la misma clave privada de la máquina. Esta relación unidireccional significa que la clave pública no puede descifrar sus propios mensajes ni descifrar nada cifrado por la clave privada.

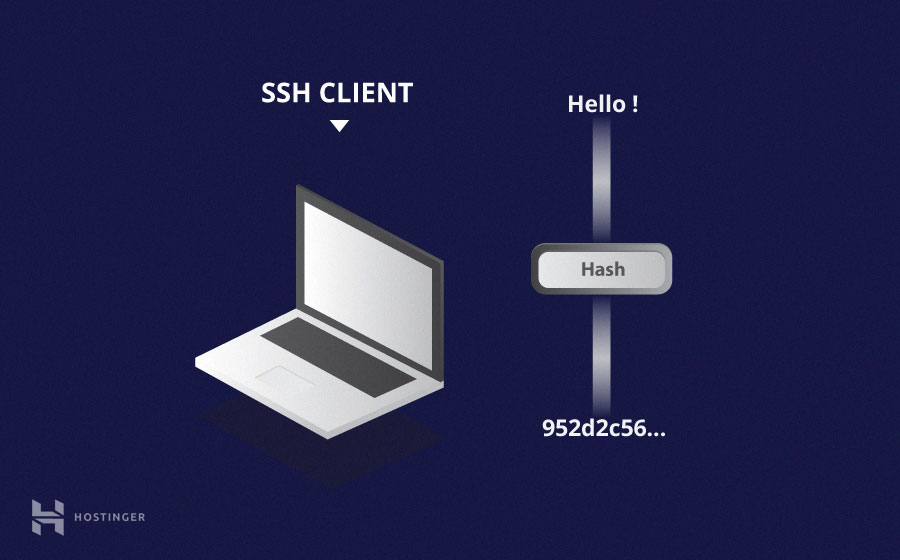
La clave privada debe permanecer privada, es decir, para que la conexión sea asegura, ningún tercero debe conocerla. La fuerza de toda la conexión reside en el hecho de que la clave privada nunca se revela, ya que es el único componente capaz de descifrar mensajes que fueron cifrados usando su propia clave pública. Por lo tanto, cualquier parte con la capacidad de descifrar mensajes firmados públicamente debe poseer la clave privada correspondiente.

A diferencia de la percepción general, el cifrado asimétrico no se utiliza para cifrar toda la sesión SSH. En lugar de eso, sólo se utiliza durante el algoritmo de intercambio de claves de cifrado simétrico. Antes de iniciar una conexión segura, ambas partes generan pares de claves públicas-privadas temporales y comparten sus respectivas claves privadas para producir la clave secreta compartida.

Una vez que se ha establecido una comunicación simétrica segura, el servidor utiliza la clave pública de los clientes para generar y desafiar y transmitirla al cliente para su autenticación. Si el cliente puede descifrar correctamente el mensaje, significa que contiene la clave privada necesaria para la conexión. Y entonces comienza la sesión SSH.

**Hashing**

El hashing unidireccional es otra forma de criptografía utilizada en Secure Shell Connections. Las funciones de hash unidireccionales difieren de las dos formas anteriores de encriptación en el sentido de que nunca están destinadas a ser descifradas. Generan un valor único de una longitud fija para cada entrada que no muestra una tendencia clara que pueda explotarse. Esto los hace prácticamente imposibles de revertir.



Es fácil generar un hash criptográfico de una entrada dada, pero imposible de generar la entrada del hash. Esto significa que si un cliente tiene la entrada correcta, pueden generar el hash criptográfico y comparar su valor para verificar si poseen la entrada correcta.

SSH utiliza hashes para verificar la autenticidad de los mensajes. Esto se hace usando HMACs, o códigos de autenticación de mensajes basados en hash. Esto asegura que el comando recibido no se altere de ninguna manera.

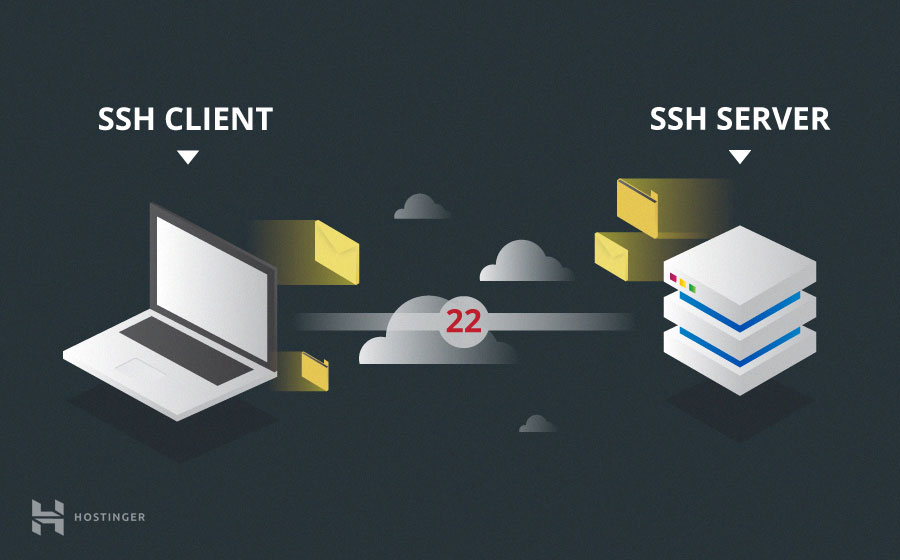
Mientras se selecciona el algoritmo de cifrado simétrico, también se selecciona un algoritmo de autenticación de mensajes adecuado (el más conocido es md5). Esto funciona de manera similar a cómo se selecciona el cifrado, como se explica en la sección de cifrado simétrico.

Todo mensaje transmitido debe contener un MAC, que se calcula utilizando la clave simétrica, el número de secuencia de paquetes y el contenido del mensaje. Se envía fuera de los datos cifrados simétricamente como la sección final del paquete de comunicaciones.

**¿Cómo funciona el protocolo SSH con estas técnicas de cifrado?**

La forma en que funciona SSH es mediante el uso de un modelo cliente-servidor para permitir la autenticación de dos sistemas remotos y el cifrado de los datos que pasa entre ellos.

SSH opera en el puerto TCP 22 de forma predeterminada (aunque esto se puede cambiar si es necesario). El host (servidor) escucha en el puerto 22 (o cualquier otro puerto SSH asignado) para las conexiones entrantes. Organiza la conexión segura mediante la autenticación del cliente y la apertura del entorno de shell correcto si la verificación tiene éxito.



El cliente debe iniciar la conexión SSH iniciando el protocolo TCP (conexión a 3 bandas) con el servidor, asegurando una conexión simétrica segura, verificando si la identidad mostrada por el servidor coincide con los registros anteriores (normalmente grabados en un archivo de almacén de claves RSA) y presenta las credenciales de usuario necesarias para autenticar la conexión.

Hay dos etapas para establecer una conexión: primero ambos sistemas deben acordar estándares de cifrado para proteger futuras comunicaciones, y segundo, el usuario debe autenticarse. Si las credenciales coinciden, se concede acceso al usuario.

**Negociación de cifrado de sesión**

Cuando un cliente intenta conectarse al servidor a través de TCP, el servidor presenta los protocolos de cifrado y las versiones respectivas que soporta. Si el cliente tiene un par similar de protocolo y versión, se alcanza un acuerdo y se inicia la conexión con el protocolo aceptado. El servidor también utiliza una clave pública asimétrica que el cliente puede utilizar para verificar la autenticidad del host.

Una vez que esto se establece, las dos partes usan lo que se conoce como Algoritmo de Intercambio de Claves Diffie-Hellman para crear una clave simétrica. Este algoritmo permite que tanto el cliente como el servidor lleguen a una clave de cifrado compartida que se utilizará en adelante para cifrar toda la sesión de comunicación. La ventaja de esto es la posibilidad de negociar nuevas claves cada cierto tiempo dificultando el posible descifrado y que si por cualquier circunstancia queda comprometida una clave privada, no se pueda descifrar toda la comunicación recibida cifrada con la parte pública de dicha clave.

Aquí es cómo el algoritmo trabaja en un nivel muy básico:

1. Tanto el cliente como el servidor coinciden en un número primo muy grande, que por supuesto no tiene ningún factor en común. Este valor de número primo también se conoce como el **valor semilla** (seed value).

2. Luego, las dos partes acuerdan un mecanismo de cifrado común para generar otro conjunto de valores manipulando los valores semilla de una manera algorítmica específica. Estos mecanismos, también conocidos como generadores de cifrado, realizan grandes operaciones sobre la semilla. Un ejemplo de dicho generador es AES (Advanced Encryption Standard).

3. Ambas partes generan independientemente otro número primo. Esto se utiliza como una clave privada secreta para la interacción.

4. Esta clave privada recién generada, con el número compartido y el algoritmo de cifrado (por ejemplo, AES), se utiliza para calcular una clave pública que se distribuye a la otra computadora.

5. A continuación, las partes utilizan su clave privada personal, la clave pública compartida de la otra máquina y el número primo original para crear una clave compartida final. Esta clave se calcula de forma independiente por ambos equipos, pero creará la misma clave de cifrado en ambos lados.

6. Ahora que ambas partes tienen una clave compartida, pueden cifrar simétricamente toda la sesión SSH. La misma clave se puede utilizar para cifrar y descifrar mensajes (leer: sección sobre cifrado simétrico).

Ahora que se ha establecido la sesión cifrada segura simétricamente, el usuario debe ser autenticado.

**Autenticación del usuario**

La etapa final antes de que se conceda al usuario acceso al servidor es la autenticación de sus credenciales. Para ello, la mayoría de los usuarios de SSH utilizan una contraseña.

Se le pide al usuario que introduzca el nombre de usuario, seguido de la contraseña. Estas credenciales pasan con seguridad a través del túnel cifrado simétricamente, así que no hay ninguna posibilidad de que sean capturadas por un tercero.

Aunque las contraseñas se cifran, todavía no se recomienda usar contraseñas para conexiones seguras. Esto se debe a que muchos robots pueden simplemente realizar ataque de fuerza bruta para descifrar contraseñas fáciles o predeterminadas y así obtener acceso a tu cuenta. En vez de eso, la alternativa recomendada es un par de claves SSH.

Se trata de un conjunto de claves asimétricas utilizadas para autenticar al usuario sin necesidad de introducir ninguna contraseña.

**Conclusión**

Obtener una comprensión en profundidad de la forma subyacente de funcionamiento de SSH puede ayudar a los usuarios a comprender los aspectos de seguridad de esta tecnología. La mayoría de la gente considera este proceso extremadamente complejo y poco comprensible, pero es mucho más simple de lo que la mayoría piensa.

Si te estás preguntando cuánto tiempo toma una computadora para calcular un hash y autenticar a un usuario, bueno, sucede en menos de un segundo. De hecho, la mayor cantidad de tiempo se gasta en la transferencia de datos a través de Internet.

Con suerte, este tutorial de SSH te ha ayudado a ver la forma en que se pueden combinar diferentes tecnologías para crear un sistema robusto en el que cada mecanismo tiene un papel muy importante que desempeñar. Además, ahora ya sabes por qué Telnet se convirtió en una cosa del pasado tan pronto apareció SSH.