SSH

Secure Shell

SSH

- A network protocol primarily for secure remote command execution.
- Requires SSH server and SSH client programs.
- Two major versions, SSH-1 and SSH-2.
- Used primarily on Linux.
- Intended as a replacement for telnet.
- Uses port 22.

Plaintext Passwords

- Programs like telnet, ftp and rsh prompt for login names and passwords.
- These are sent as plaintext over the network.
- And therefore vulnerable to TCP/IP packet sniffing.
- Also any commands you give or files you upload and download are not encrypted.

ssh-keygen

- ssh-keygen generates a public key private key pair for you.
- You will be prompted for a passphrase.
- By default your
  - private key is stored in .ssh/id_rsa
  - public key is stored in .ssh/id_rsa.pub
- (on the client machine)

ssh-keygen

- Your private key is only readable by yourself
  - -rw- --- ---
- Your public key is public
  - -rw- r-- r--

Key Based Authentication

- In order to enable key based authentication, the content of id_rsa.pub must be added to ~/.ssh/authorized_keys on the server.
- Now when you login to the SSH server, you wont be asked for a password.
- The SSH client program will read your private key from .ssh/id_rsa (on the client machine) and use that to authenticate you with the server.

SSH

- On Unix-like systems, the list of authorized keys is stored in the home folder of the user.
- In the file ~/.ssh/authorized_keys.
- When the public key is present on the server and the matching private key is present on the client, typing in the password is no longer required.

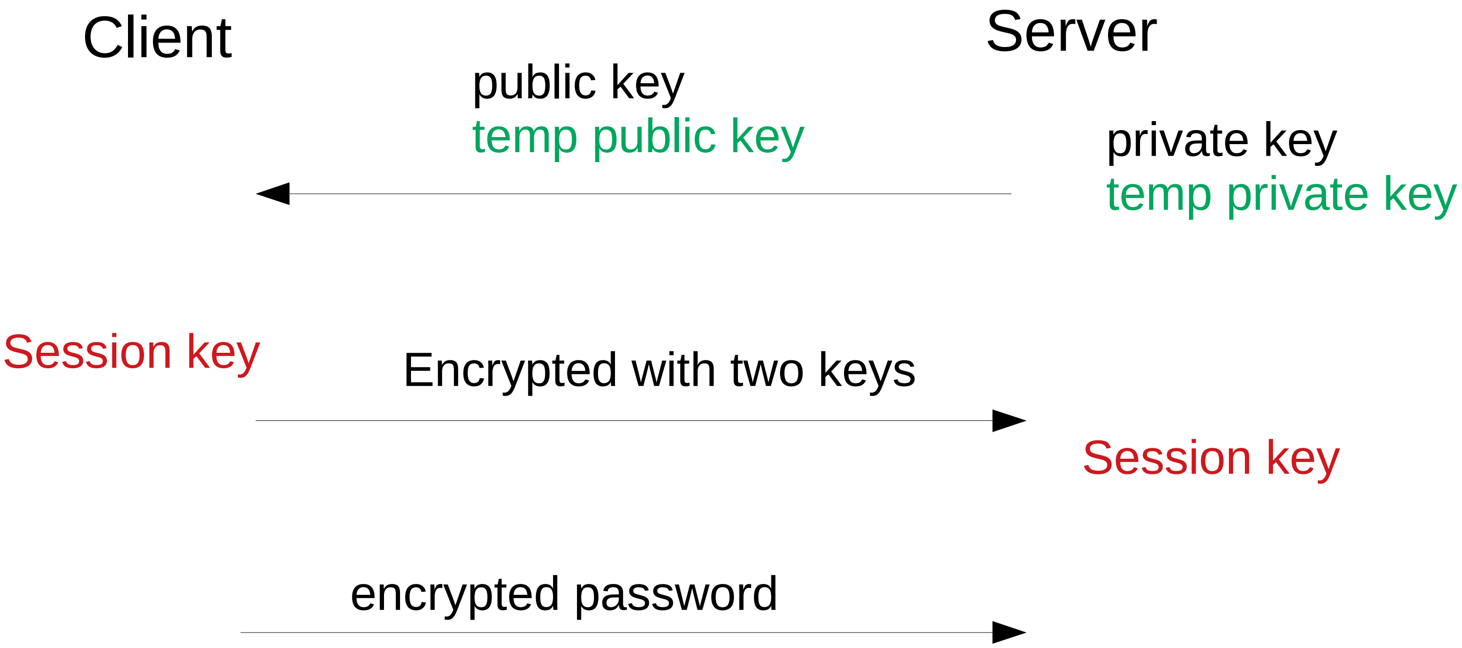
Establishing a Secure Connection

- Client connects to the server.
- The Server sends its public key (the host key) and also another temporary public key (the server key).
- [The public key is checked against ~/.ssh/authorized_keys]

Establishing a Secure Connection

- The client chooses a symmetric key (the session key).
- It encrypts it with both the public key and the temporary public key and sends it to the server.
- Both sides turn on encryption (start using the symmetric key.)

SSH with password



Client

Server

public key
temp public key

private key
temp private key

Session key

Encrypted with two keys

Session key

encrypted password

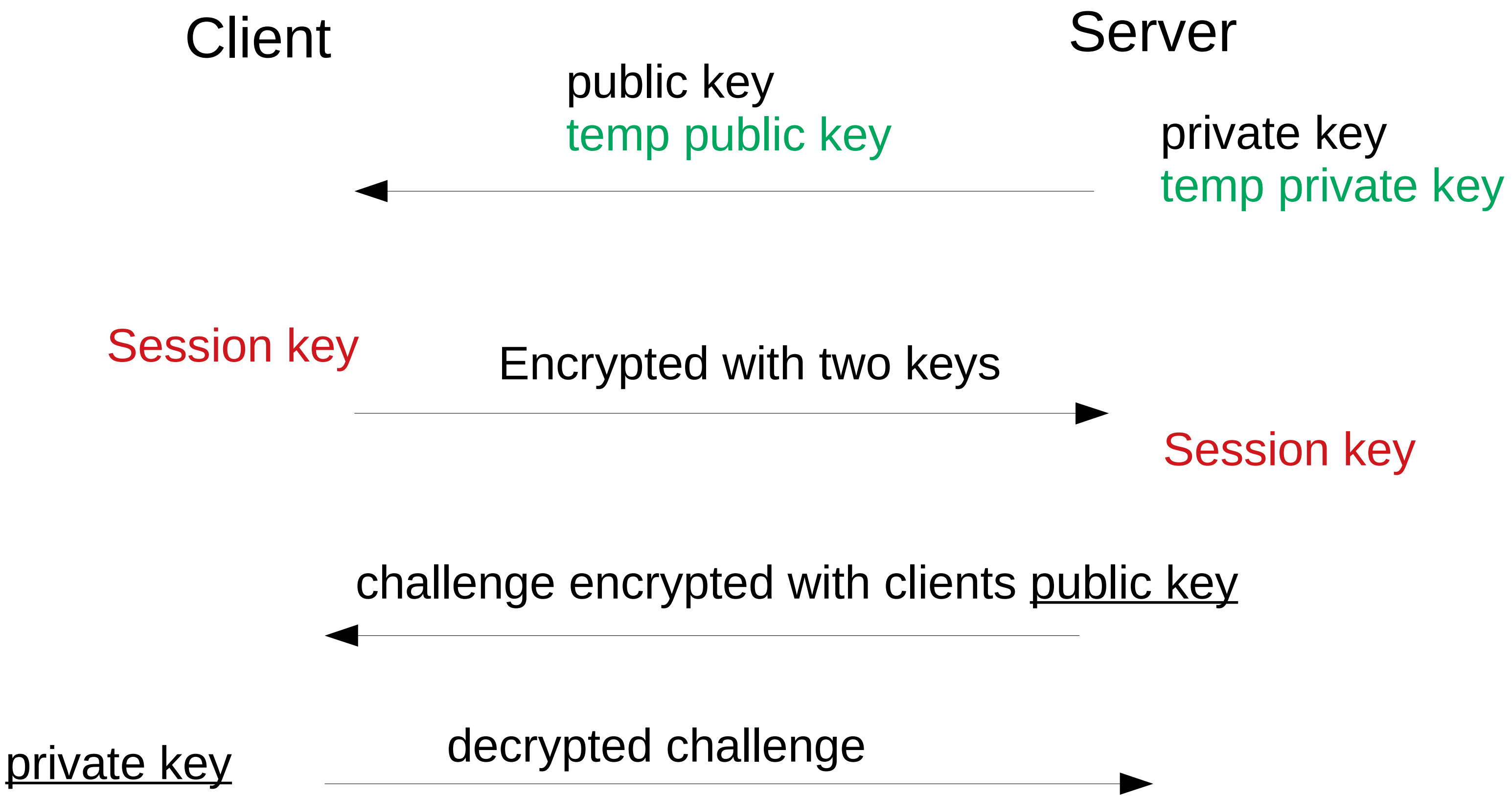
Establishing a Secure Connection

- The client now authenticates itself using either
  - password
  - PKI
- To use a password, the password is encrypted and sent to the server.

Establishing a Secure Connection (cont)

- Using PKI
- The server generates a random 256-bit string as a challenge.
- Encrypts it with the client's public key, and sends this to the client.
- The client receives the challenge and decrypts it with the corresponding private key.

SSH with PKI Authentication



Client

Server

public key
temp public key

private key
temp private key

Session key

Encrypted with two keys

Session key

challenge encrypted with clients public key

private key

decrypted challenge

Perfect Forward Secrecy

- Encrypting the session key a second time with the server key provides a property called perfect forward secrecy.
- Suppose the server was compromised and the servers private key obtained.
- Then all (recorded) sessions in the past could be decrypted.
- The use of a second server key means that old sessions would not be compromised.