Beginning Custom Projects with Raspberry Pi

Easy Authentication with SSH Keys

SSH Authentication

Many authentication methods available, commonly

- Password
- SSH keys

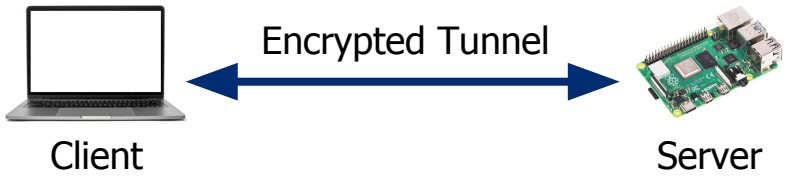
In this video

- Review password operation
- How do SSH keys differ?
- How do I set up SSH keys on my Raspberry Pi?

Result: Securely log in with no passwords!

How do passwords work? Step 1.

Step 1: Establish a secure encrypted tunnel (no auth yet)



A key exchange algorithm (called Diffie-Hellman) is used to established a "*shared secret*" (key).

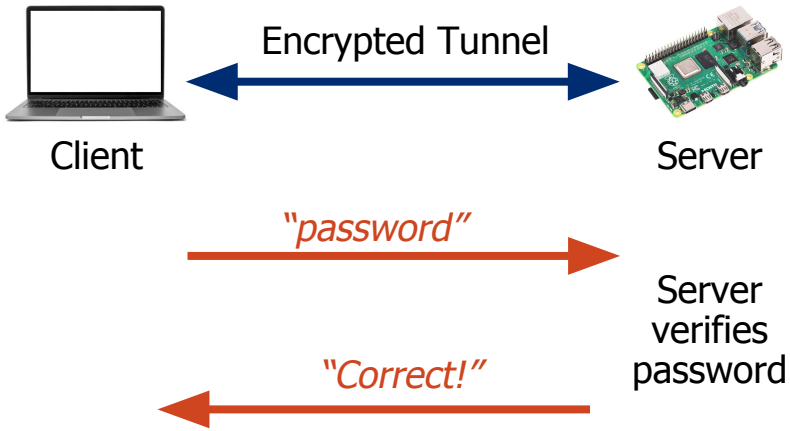
This key is used by each party to encrypt/decrypt.

When everyone uses the same key, called "*symmetric key*" cryptography

Note: Authentication has not happened yet!

How do passwords work? Step 2.

Step 2: Authenticate the user by verifying their password.

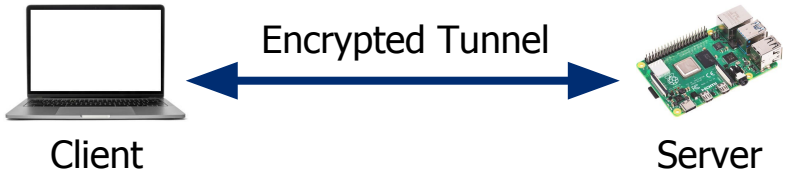


User is authenticated and can log in.

Simple, provides some security, rather inconvenient.

How Do SSH Keys Work?

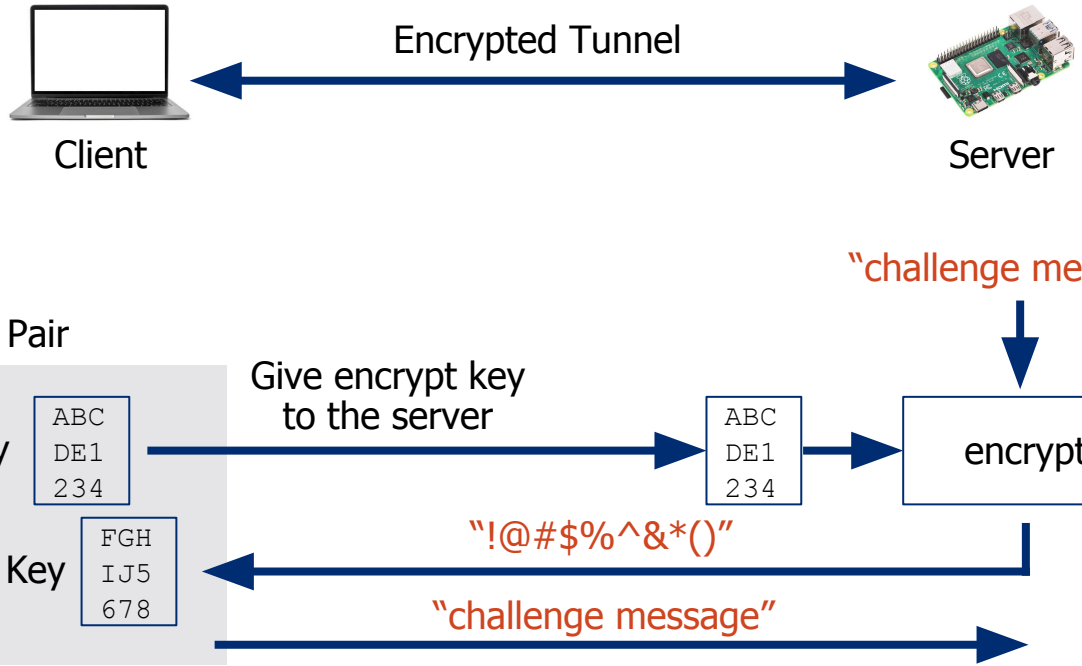
Step 1: Establish a secure encrypted tunnel (unchanged)



Step 2: Verify a challenge message encrypted with public-key authentication.

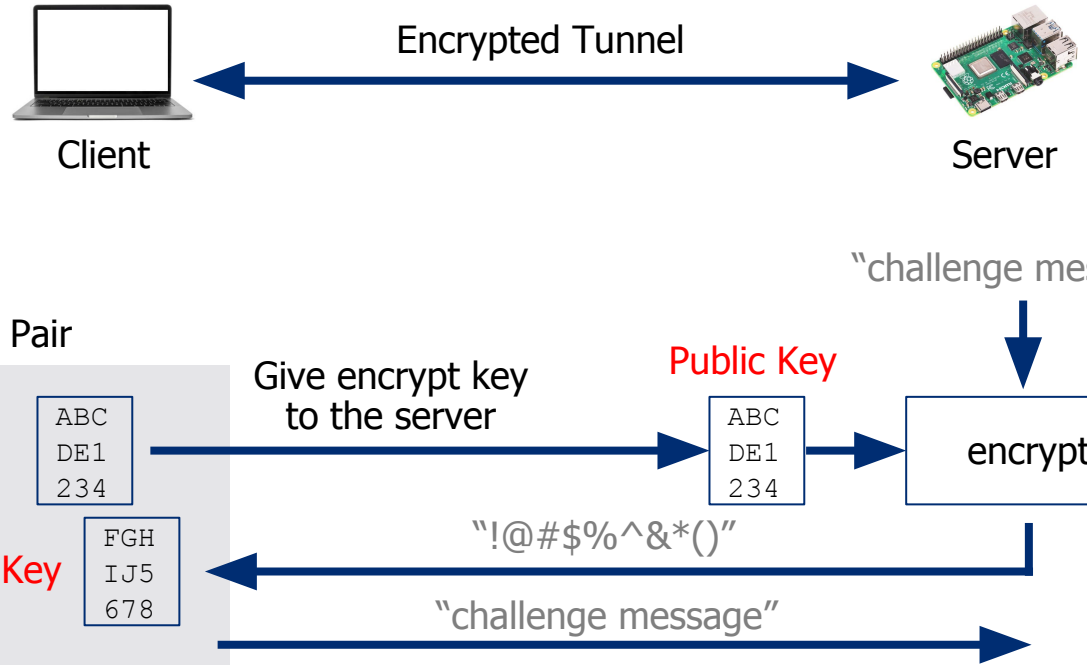
How does public-key authentication work?

Public-Key Authentication



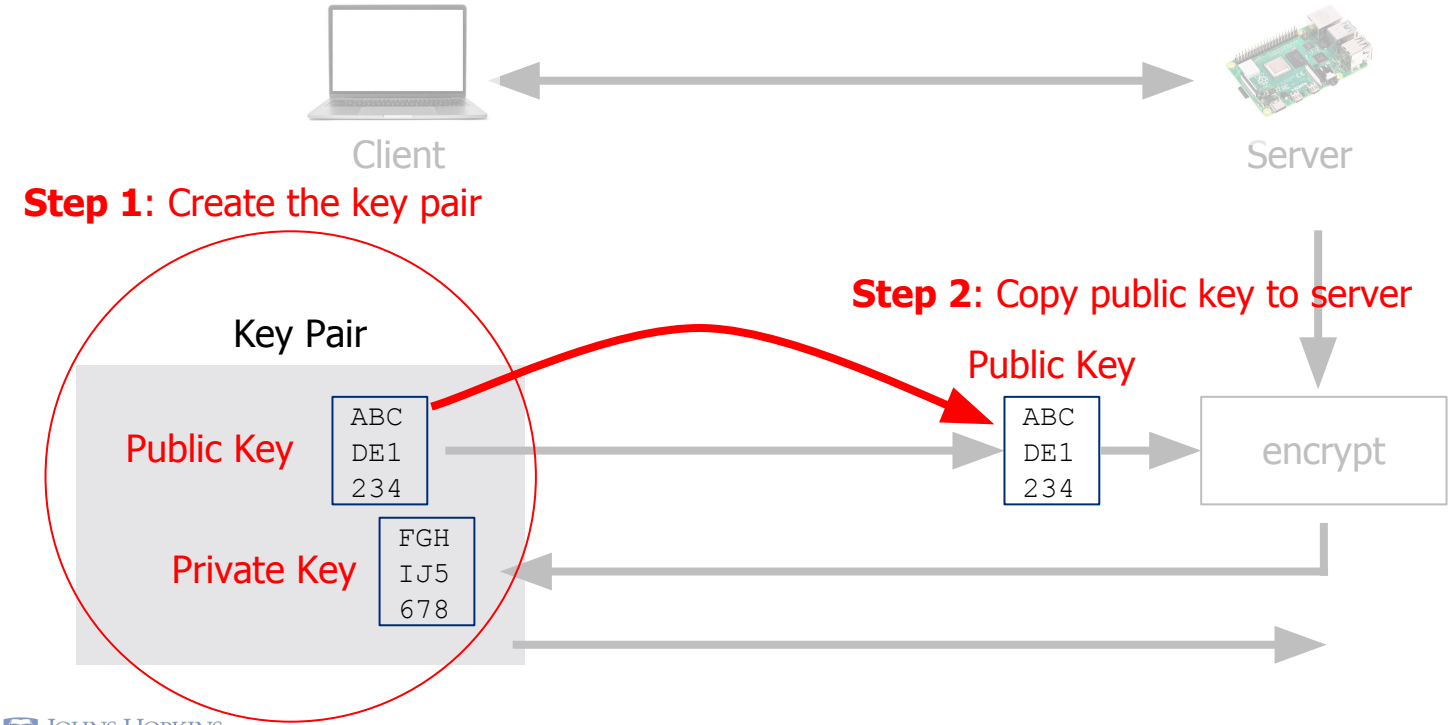
If the user is in possession of the matching "decrypt key" then we trust them.

Public-Key Terminology



If the user is in possession of the matching "private key" then we trust them.

Configuring SSH Keys



Summary

- A lot of theory, but very easy to implement.
- How to configure SSH keys:

```
$ ssh-keygen
$ scp ~/.ssh/id_rsa.pub <ip>:~/.ssh/authorized_keys
```

- Like everything we do:

  Understand, don't *memorize*

Next: Let's log in (with SSH auth!) and create our API server like we did in Module 2, but this time by using the Flask framework.

© The Johns Hopkins University 2021, All Rights Reserved.