

# **AST1501 - Introduction to Research**

**Jo Bovy**

# Intro to computing II

# Overview

- `git`
- GitHub
- SSH
- Python environments



git

# Why use Version control?

- Version control keeps the history of changes to your code (or documents, images, etc., but text files work best), allowing you to trace changes over time. This frees you from having to track changes manually.
- Most version control systems use a *central location* for the main copy of your code, which acts as
  - A crucial back-up of your work
  - A central place to share your code with yourself (for use on multiple machines) and others (e.g., collaborators)
- Branches allow you to keep multiple in-progress versions of your code that can be developed in-parallel and merged later

# git version control

- `git` is the latest and greatest version control system, probably the only one you've heard of
- `git` has a decentralized approach to version control: by default, each version ("clone") of the code *repository* has the *full history of changes*
- `git` is now closely associated with GitHub although technically they are independent from each other

# Quick `git` intro

- We will now run through some basic and advanced features of `git`, based on the prior knowledge of the course participants

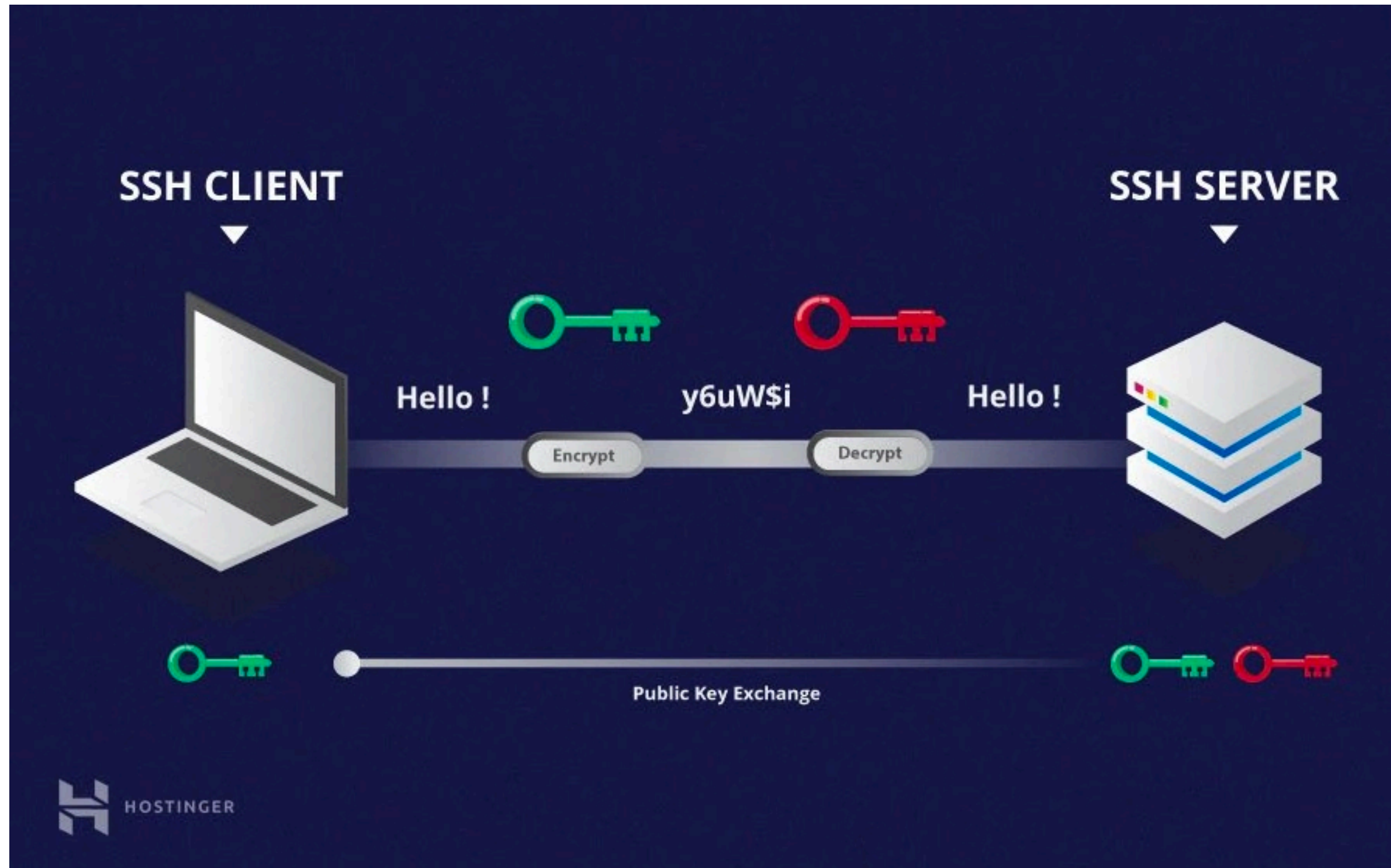


**GitHub**



# GitHub

- Tour of GitHub
- Education benefits: <https://education.github.com/benefits>



# SSH

# SSH

## Secure Shell

- Safely connecting to remote servers is done using SSH (Secure Shell), a networking protocol for computers
- Basic login

```
ssh bovy@lepus.astro.utoronto.ca
```

which will prompt you for your password

- Then get to a shell that is exactly like running a terminal on your own machine
- SSH also backs services like `rsync` to transfer files between computers

# SSH

## Keys

- Typing in your password all the time gets annoying, so better to use *SSH keys*
- SSH keys: private/public encryption key pair to authenticate as *you*
- Encryption keys work by placing your public key on, say, the server you want to log into and keeping the private key on your own machine
  - Public key can be easily matched to a private key, without allowing the private key to be determined from the public key  
(e.g., public key is large number that is the product of two large primes, private key is those two large primes)
  - **Never** share your private key; if it gets compromised, delete the pair and start over

# SSH

## Keys

- Create public/private pair using

```
ssh-keygen -t rsa
```

- Reports

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/Users/bovy/.ssh/id_rsa):
```

- Then asks

```
Enter passphrase (empty for no passphrase):
```

- Use a passphrase!

- Then get

```
Your identification has been saved in /Users/bovy/.ssh/KEYNAME
```

```
Your public key has been saved in /Users/bovy/.ssh/KEYNAME.pub
```

# SSH

## Keys

- If you need keys for multiple purposes, best to generate separate pairs in case they get compromised
- For example, I login automatically to the astro server from GitHub and I use a special SSH key for that
  - Note that for automated logins, you need to use a passphraseless key, another good reason to use separate keys!
- But when you move to a new computer, can copy private key to keep all logins working
- Passphrase will generally be remembered by your local password manager if you want

# SSH

## Configuration

- SSH configuration lives in the `.ssh` directory in your home directory
- Can contain a config file to configure logins, e.g.

```
Host NICKNAME
    Hostname SERVER.astro.utoronto.ca
    User USERNAME
    ForwardAgent yes
```

- Then you can simply login with

```
ssh NICKNAME
```

- Or `rsync`

```
rsync -azv NICKNAME:/some/directory .
```

# SSH

## Remote configuration

- To be able to login using SSH, you need to add your *public* key to the `authorized_keys` file in the `.ssh` directory on the remote machine
- For example

```
cat ~/.ssh/id_rsa.pub | ssh -l USERNAME  
SERVER.astro.utoronto.ca 'sh -c "cat - >> ~/.ssh/  
authorized_keys"'
```

Where `id_rsa.pub` is your public key

- Note that you can also use an extension in VS code to interact with remote code locally (e.g., locally edit Jupyter notebooks that are running remotely)