CMPS 4663 CRYPTO

IMPLEMENTING OUR OWN PUBLIC KEY ENCRYPTION LAYER

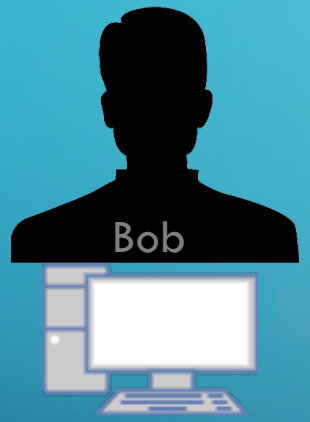
TERMS

Just a few

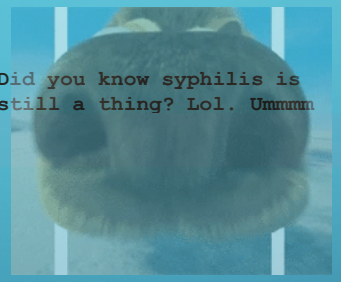
- **Alice** – Message sender or Receiver
- **Bob** – Same as Alice
  - Usually Alice send to Bob (A to B get it!).
- **Eve** – Eavesdropper (assumed to be intercepting signal , packets, whatever)
- **Packet Sniffer** – Something that grabs and logs every packet sent on a network. This is a pretty easy task (google wireshark or snort).

INTRODUCTION

Oversimplified Symmetric Key Example



Bob wants to send a message to Alice



Eve employs her packet sniffer.

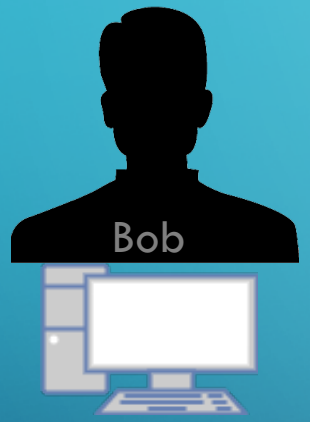
JUST ENCRYPT

Oversimplified Symmetric Key Example

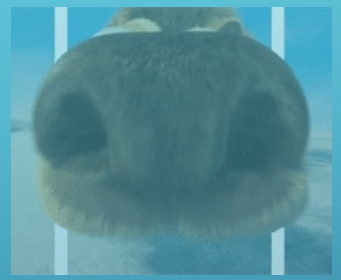
Bob wants to send a message to Alice

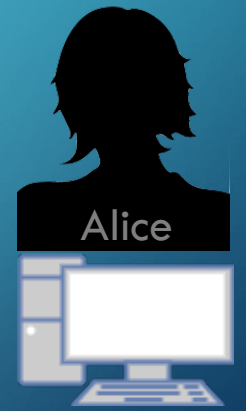
SEND A KEY

Oversimplified Symmetric Key Example



Bob wants to send a message to Alice

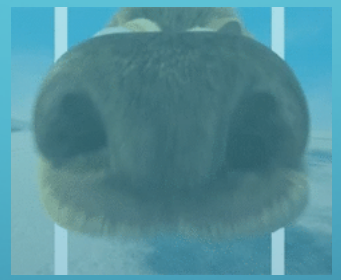


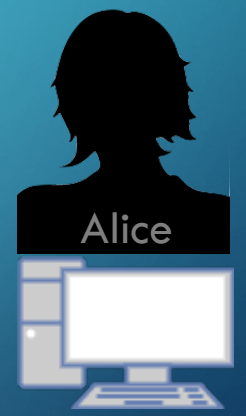


SEND A KEY

Oversimplified Symmetric Key Example

Bob wants to send a message to Alice



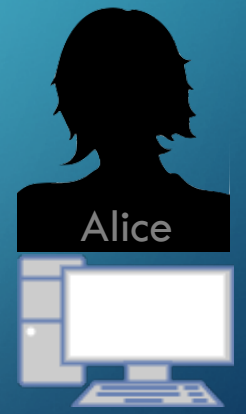


SEND A KEY

Oversimplified Symmetric Key Example

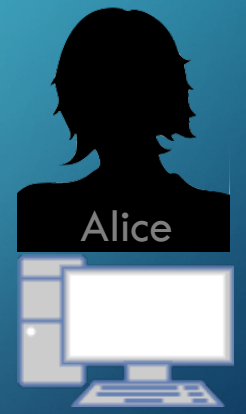
RESOLUTION?

Assume there is always an EVE.

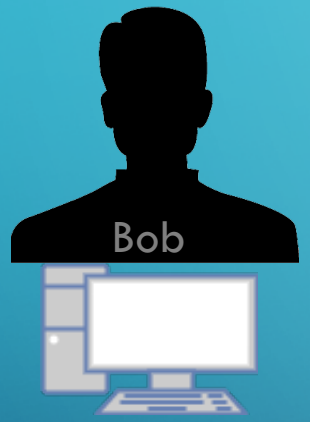


RESOLUTION?

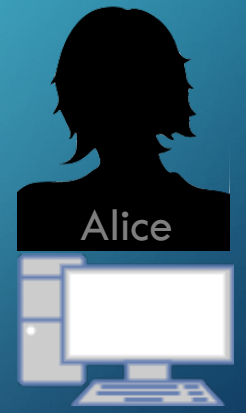
Use public key encryption!



PUBLIC KEY



Bob wants to send a message to Alice

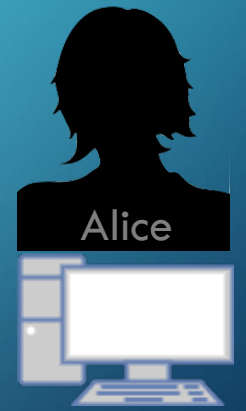


We assume Eve is always listening!

PUBLIC KEY

Generate Keys

Bob and Alice Generate Keys



Public

Private

Public

Private

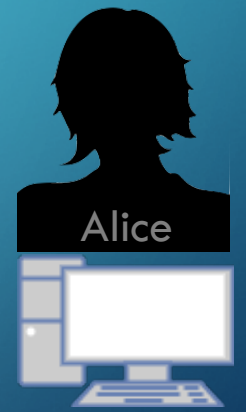
PUBLIC KEY

Swap Keys

Bob and Alice Swap Public Keys

Public

Private



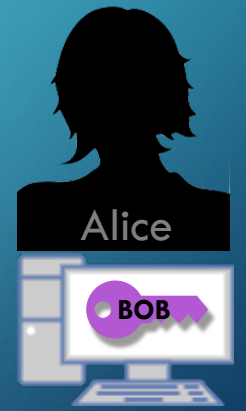
Public

Private

PUBLIC KEY

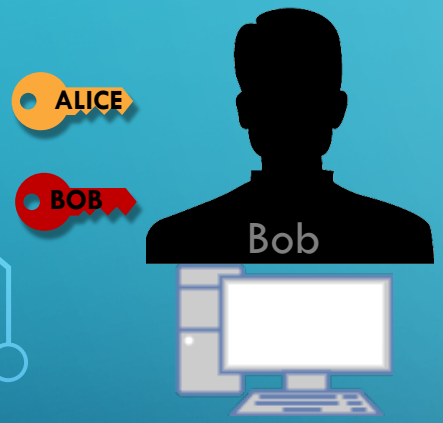
Swap Keys

They save their own keys, and each others public key.

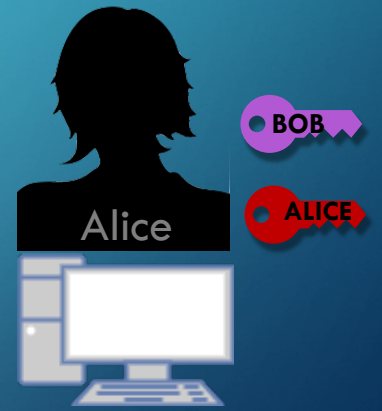


PUBLIC KEY

Save Each Others Public Key



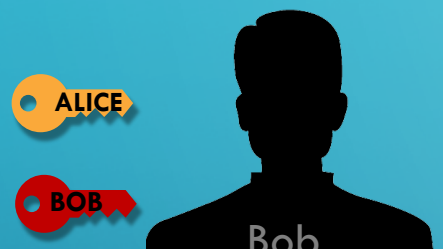
Now when Bob wants to send Alice a message
He encrypts it with her **public key**

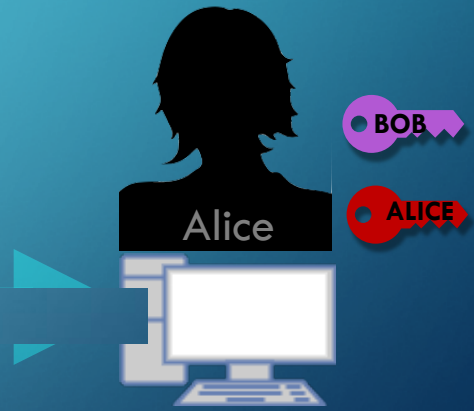


PUBLIC KEY

Use Public Key to Encrypt

Now when Bob wants to send Alice a message
He encrypts it with her **public key**



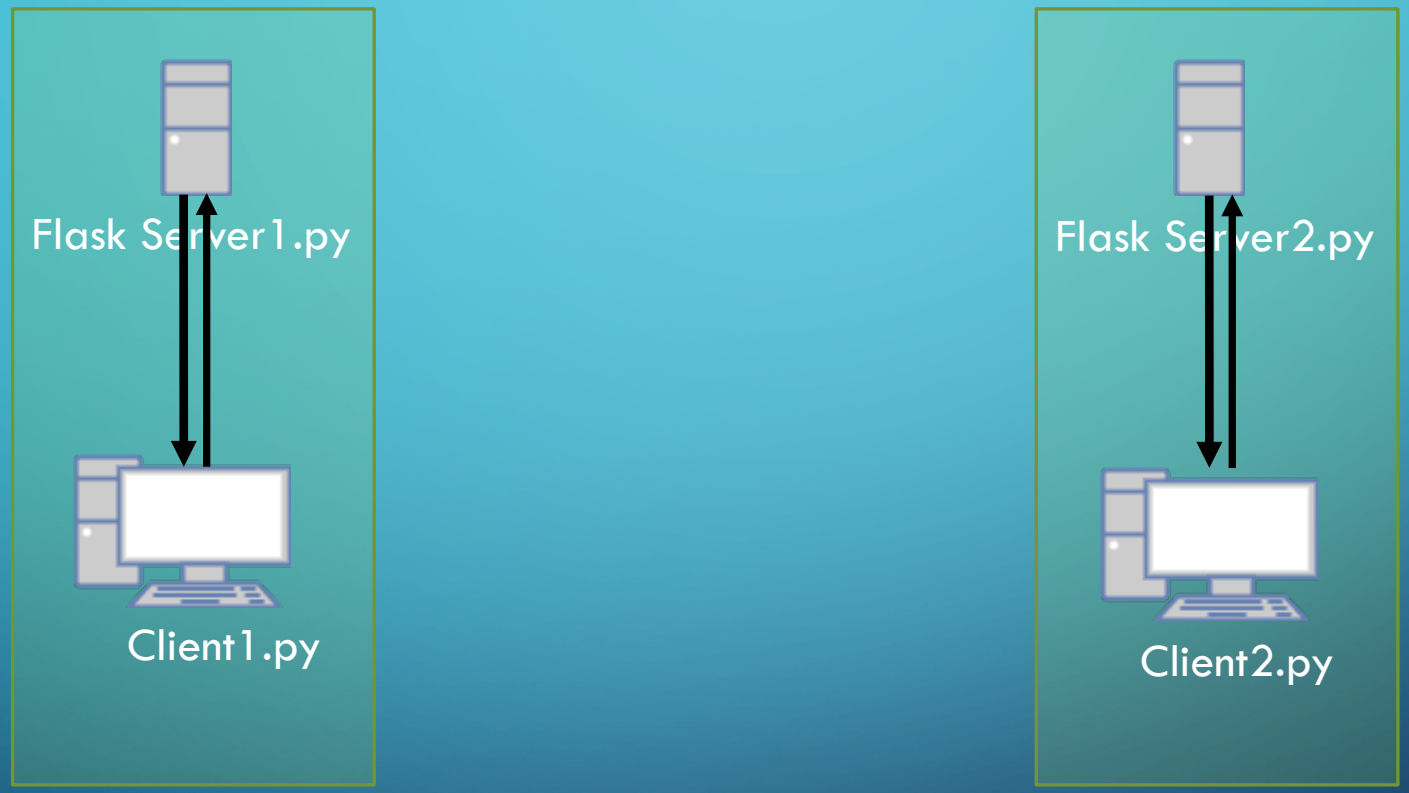


PUBLIC KEY

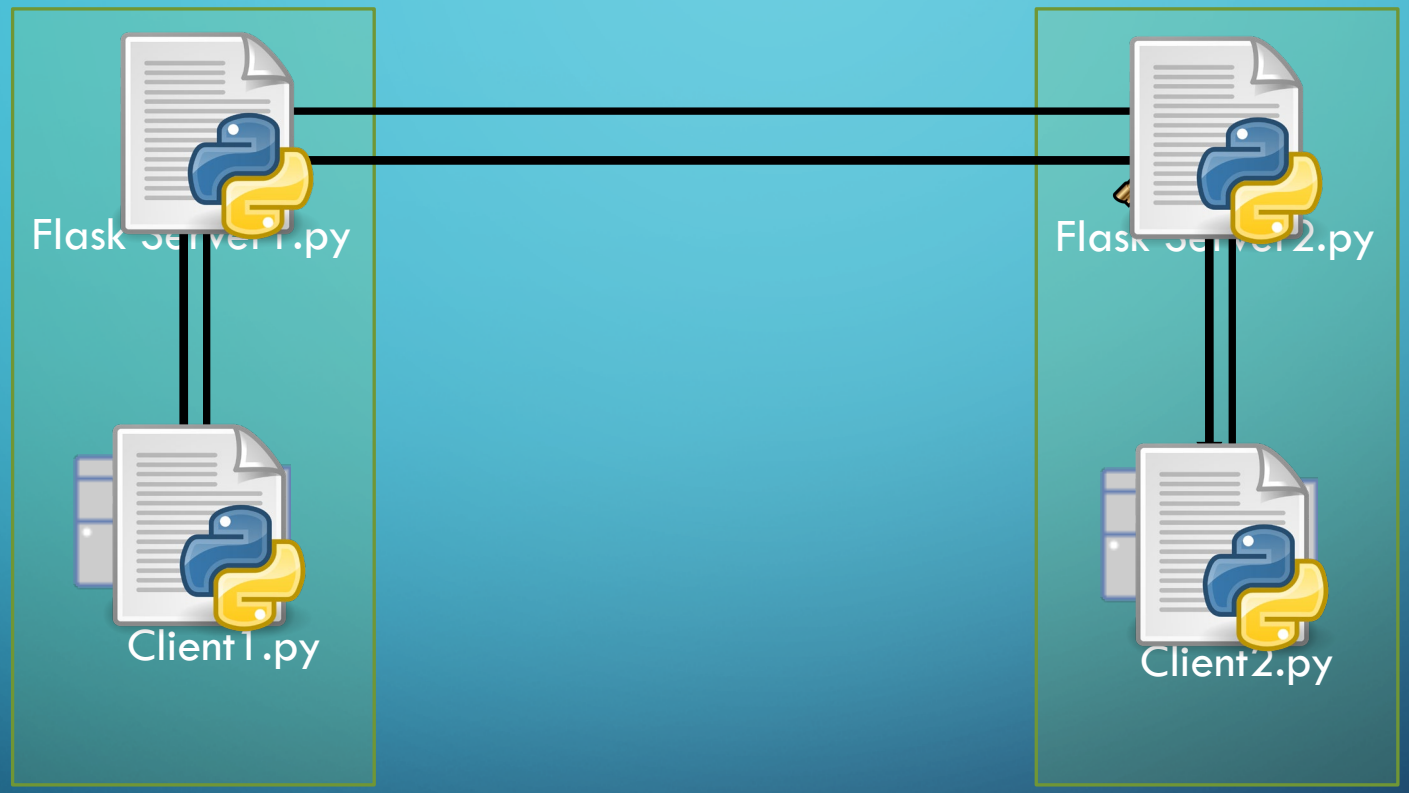
Use Private Key to Decrypt

When Alice receives the message, she decrypts it with her **private key**

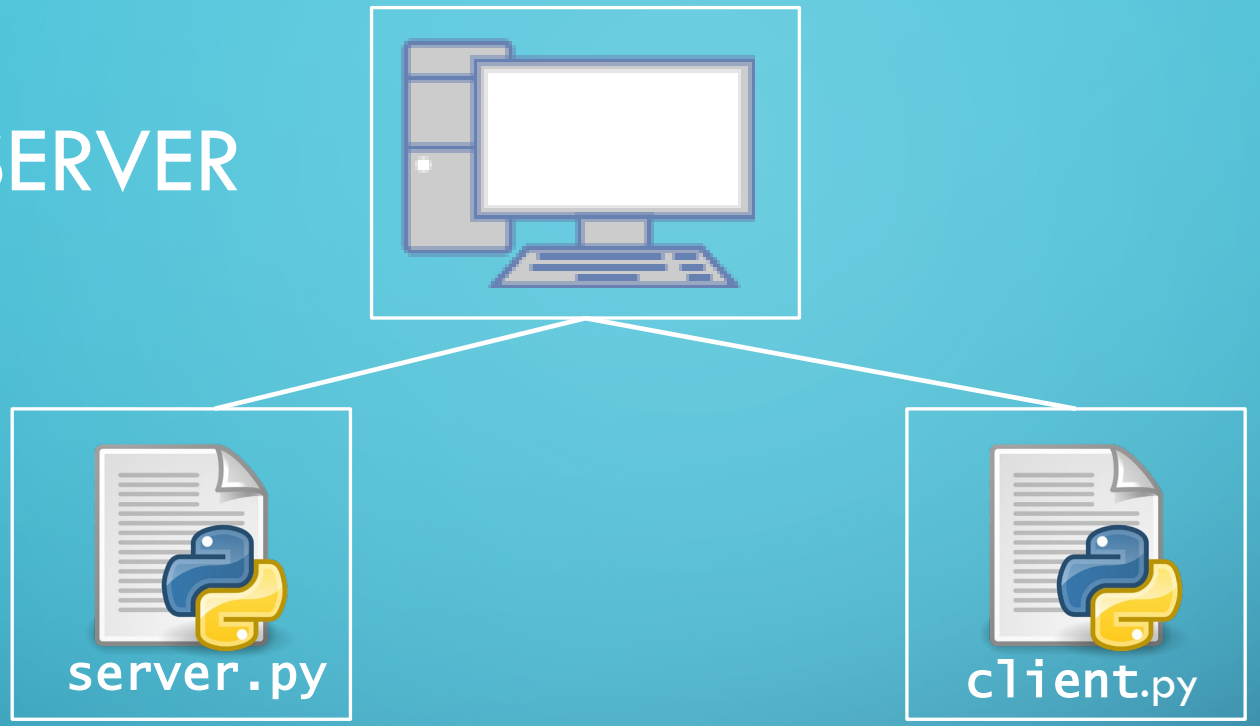
CLIENT SERVER



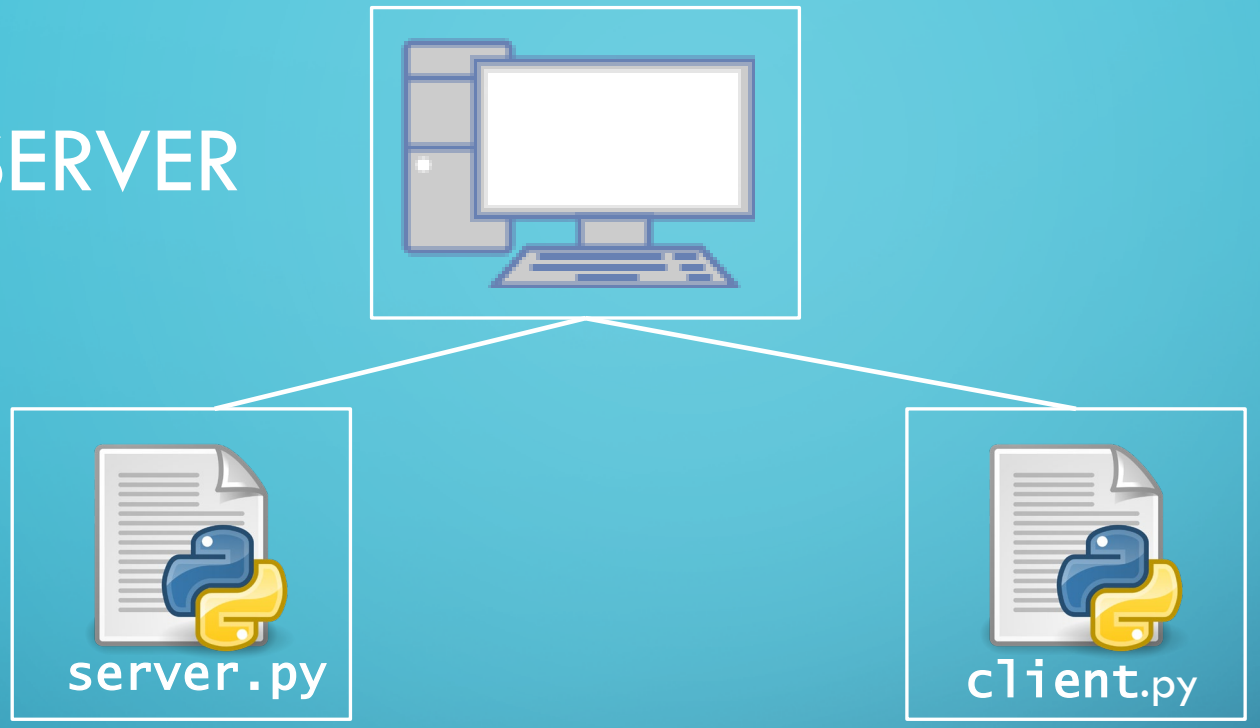
CLIENT SERVER



CLIENT SERVER

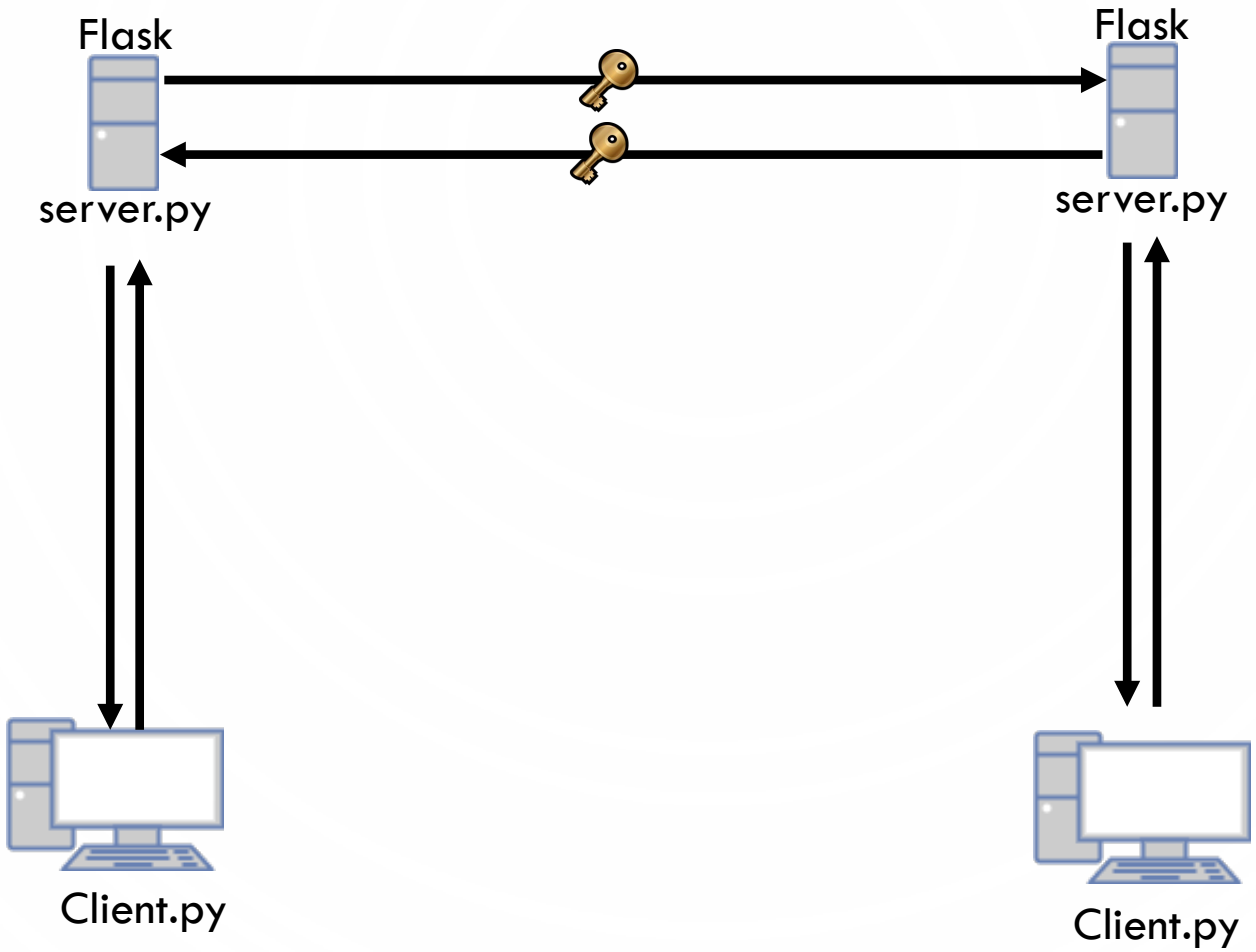


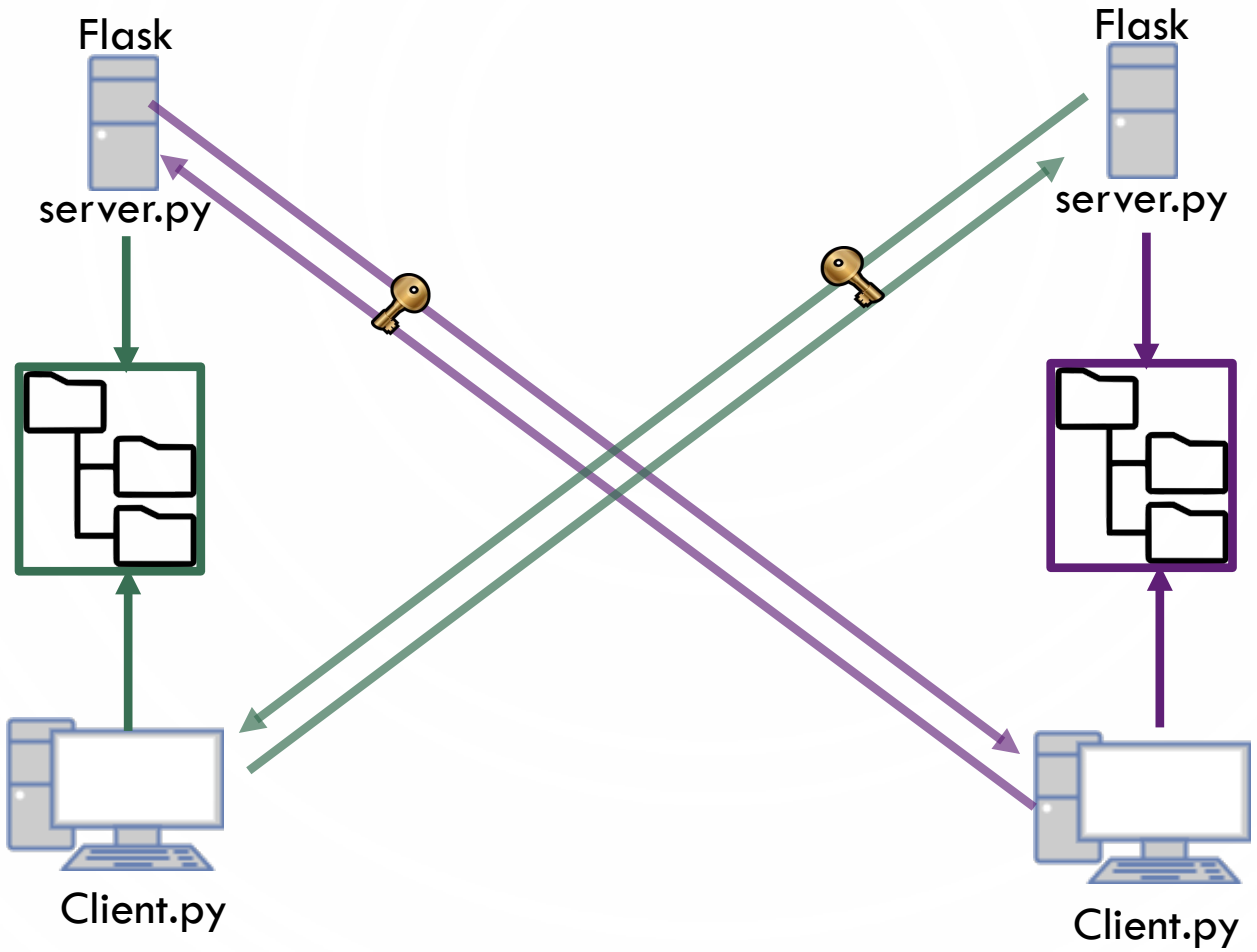
CLIENT SERVER



- Our connection to “outside” world (other computers)
- Listens on port for GET and POST requests
- No interactive console

- No connection to outside world
- Our local interface to “server”
- Provides interactive console


Flask
server.py
Flask
server.py
Client.py
Client.py


Flask
server.py
Flask
server.py
Client.py
Client.py