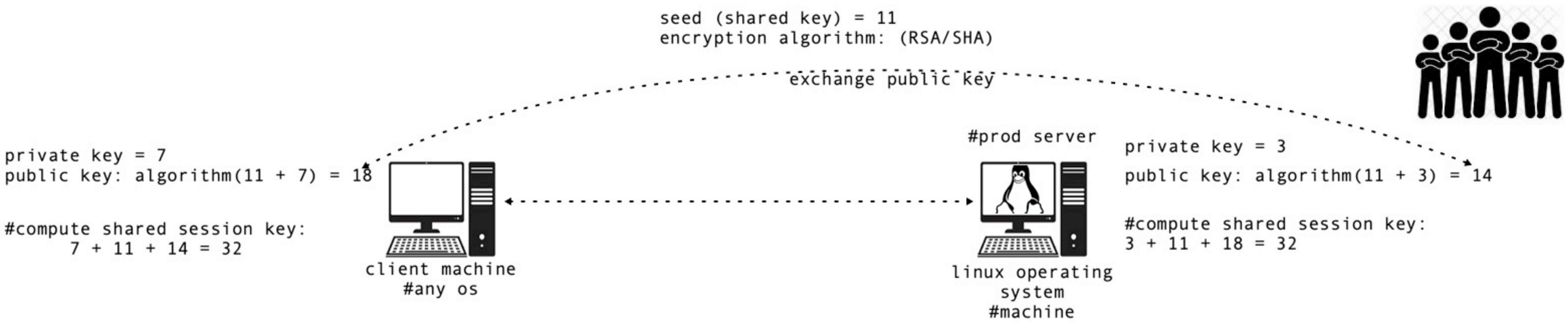
How does SSH works?
To ensure all the communication that takes place between the client and the server is secured, SSH establishes the connection and communicates with the remote computer in #2 stages
1. session establishment
2. authentication

1. How does the session would be established between both parties in initializing the communication?
The client computer sends an request to the server computer asking for starting an SSH session for communicating with each other. The server computer would accept the request, so that the session establishment would begin



To ensure the communication between these 2 computers to be secured, they need to use encryption technic for exchanging the data. For this both these parties should agree upon an symmetric key to make the communication secure. The key should not be generated by one party and share with another party, because during the key exchange process anyone can steal the key and makes the communication in-secure.

Both the parties will compute the symmetric key individually and arrives to the same key based on sequence of operations as below.
1. both the parties agrees on a large prime number, which will be served as a seed value (shared prime number)
2. both the parties agrees on an encryption algorithm like AES, RSA, SHA etc which is used for encrypting the data
3. Independently each party comes up with another prime number which is kept secret from other party (this is considered as private key)
4. generated private key (private prime number), encryption algorithm and shared prime number are used to generate an public key that is derived from their private key
5. both the public keys are exchanged between the parties
6. upon receiving the public keys of each other:
  a) by using the private prime number (private key)
  b) public key of other person
  c) shared prime number
  generate an session key, where both the parties would endup in computing the same key, this key that is computed by both the parties is called “shared session key”
7. by using the shared session key both the parties will encrypt and exchange the data between them

---

#2. authentication
upon computing the shared session key (symmetric key) by both the parties to encrypt and exchange the data, the server challenges the client asking him to authenticate himself to grant access to the server machine

The client inorder to establish the communication or gain access, he has to send the username/password of the linux user using which he want to perform the operations on the server machine. In order to send the username/password of the linux user, the client will encrypt using the shared session key and would send to the server

The server upon receiving the encrypted username/password, will decrypt using the shared session key and validate the credentials. If the username/password is invalid, it denies the access to the client computer.

If valid grants the access allowing the remote client machine to communicate with the server machine onbehalf of the linux user he has been authenticated and exchange the data using the shared computed session key.

This process of authenticating the client by asking username/password in gaining the access to the remote server computer is called “Password based authentication” technic.

The “password authenticiation” technic in authenticating the client in gaining the access to the remote computer has few dis-advantages:
1. exchanging the username/password over the network poses security risk, no matter which encryption technic we used
2. By allowing the client to connect to the server using username/password, grants permanent access to the remote computer. In future if we want to revoke the access to the client, it would not possible or might lead to challenges

So it is advices not to use “passwordAuthentication” in authenticating and granting the access to the remote computer.

How to overcome the above problem in granting the access to the server?
That is where “Asymmetric encryption keys” are used for authenticating the client and granting the access to the remote machine.