



UNLV CYBER SECURITY CLUB

Tutorial: PGP Key Creation & SSH Key

Overview

All slides can be found at

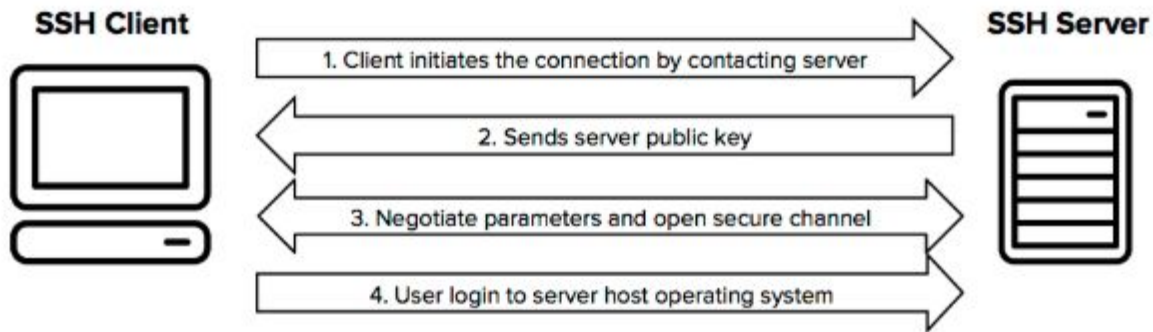
*[https://github.com/layer-zero-unlv/
training-sessions](https://github.com/layer-zero-unlv/training-sessions)*

- History
 - Asymmetric Key Encryption
 - SSH Key Creation
 - PGP Key Creation
-

History

SSH

- Secure Shell Protocol (Tatu Ylonen 1995)
- Port 22



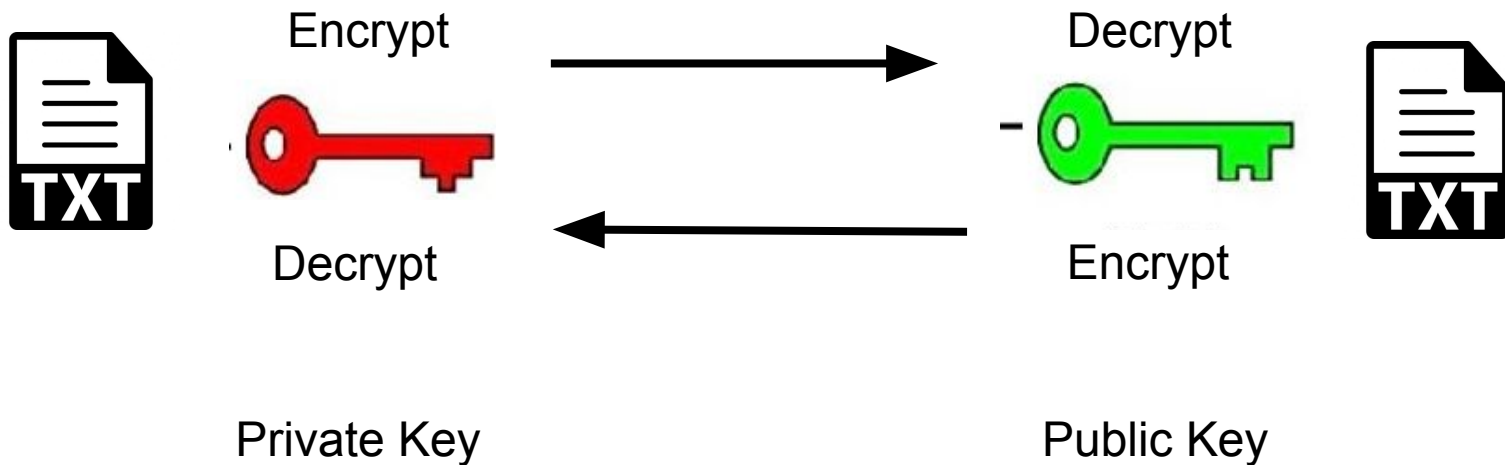
Pretty Good Privacy (PGP)

- Most widely used encryption standard for end-to-end encryption
 - Banks, healthcare, high regulated industries etc....
- Encryption program - Open PGP
 - GnuPG - GNU Implementation
 - Hashing, data compression, symmetric/asymmetric key cryptography
 - Key Management
- End to End Encryption Email
 - Protonmail
 - <https://protonmail.com/>

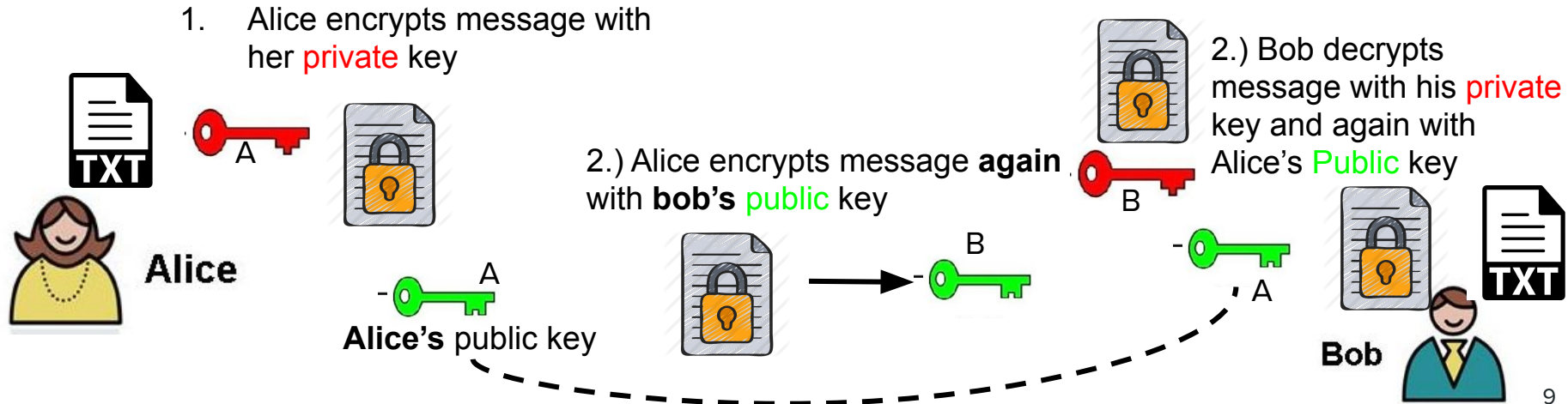
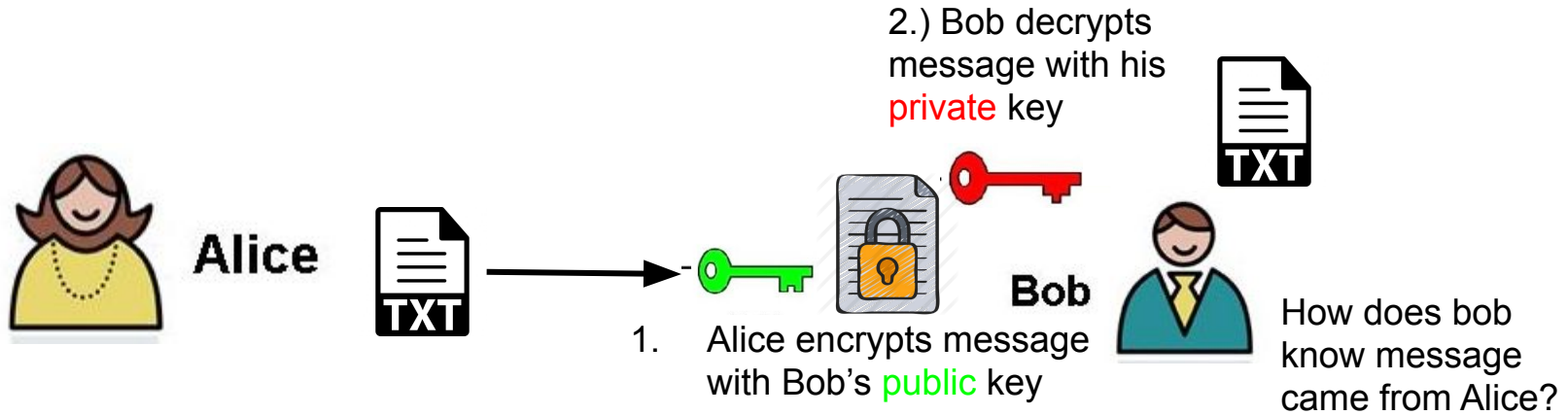
Asymmetric Key Encryption

RSA (Rivest–Shamir–Adleman)

- Asymmetric Key Encryption
- Public and Private Keys
- The keys can encrypt and decrypt each other!



RSA



Pretty Good Privacy (PGP)

Gnupg: Create a GPG Key

Check if it's already installed

```
~ $ which gpg
```

If not, run the command to install

```
~ $ sudo apt install gnupg
```

Create your key

```
~ $ gpg --gen-key
```

```
~ $ gpg --full-generate-key
```

GnuPG: Encrypt a Message for Yourself

Change directory to your desktop (so we can see the files easier)

```
~ $ cd ~/Desktop (mac/linux)
```

```
~ $ cd Desktop\ (windows)
```

Create a message

```
~ $ touch message.txt && echo "Hello" > message.txt
```

Encrypt the message to yourself. (use `gpg --help` to know what the letters are)

```
~ $ gpg -e -a -r <email> <file>
```

Inspect the file

```
~ $ cat message.txt.asc
```

GnuPG: Decrypt the Message From Yourself

Decrypt the message

```
~ $ sudo gpg -o output.txt -d message.txt.asc
```

View the plain text message

```
~ $ cat output.txt
```

GnuPG: Export your Public Key

Export your public key

```
~ $ gpg -a --export <email> > pub-key.asc
```

View your public key

```
~ $ cat pub-key.asc
```

Now email pub-key.asc file to a friend

GnuPG: Import a Public Key

Download the pub-key.asc (sent from your friend) to your desktop

Import key

```
~ $ gpg --import pub-key.asc
```

View your public keys in your key ring

```
~ $ gpg --list-keys
```

GnuPG: Encrypt a message with your friends Key

Encrypt Message

```
~ $ gpg -a -e -r <friends_email> <file>
```

Try decrypting it and see what happens....(it won't work)

Encrypt a message with your key too!

```
~ $ gpg -a -e -r <friends_email> -r <your_email> <file>
```

Decrypt the message sent from your friend again...it works!

```
~ $ gpg -d <file>
```


GnuPG: Decrypt a message

Email the encrypted message to your friend for them to decrypt

Decrypt the message sent from your friend

```
~ $ gpg -d <file>
```

SSH Key Creation

Installing OpenSSH

Check if it's already installed

```
~ $ sudo systemctl status ssh
```

If not running, need to instal OpenSSH

```
~ $ sudo apt openssh-server
```

Check if the server is running

```
~ $ sudo systemctl status ssh
```

Creating the SSH Key

Ensure ssh-keygen is installed

```
~ $ which ssh-keygen
```

Create the SSH using RSA encryption

```
~ $ ssh-keygen -t rsa
```

Answer the following questions

```
Enter file in which to save the key  
(/home/demo/.ssh/id_rsa)
```

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

Creating the SSH Key

Ensure key was created

```
~ $ cd ~/demo/.ssh/
```

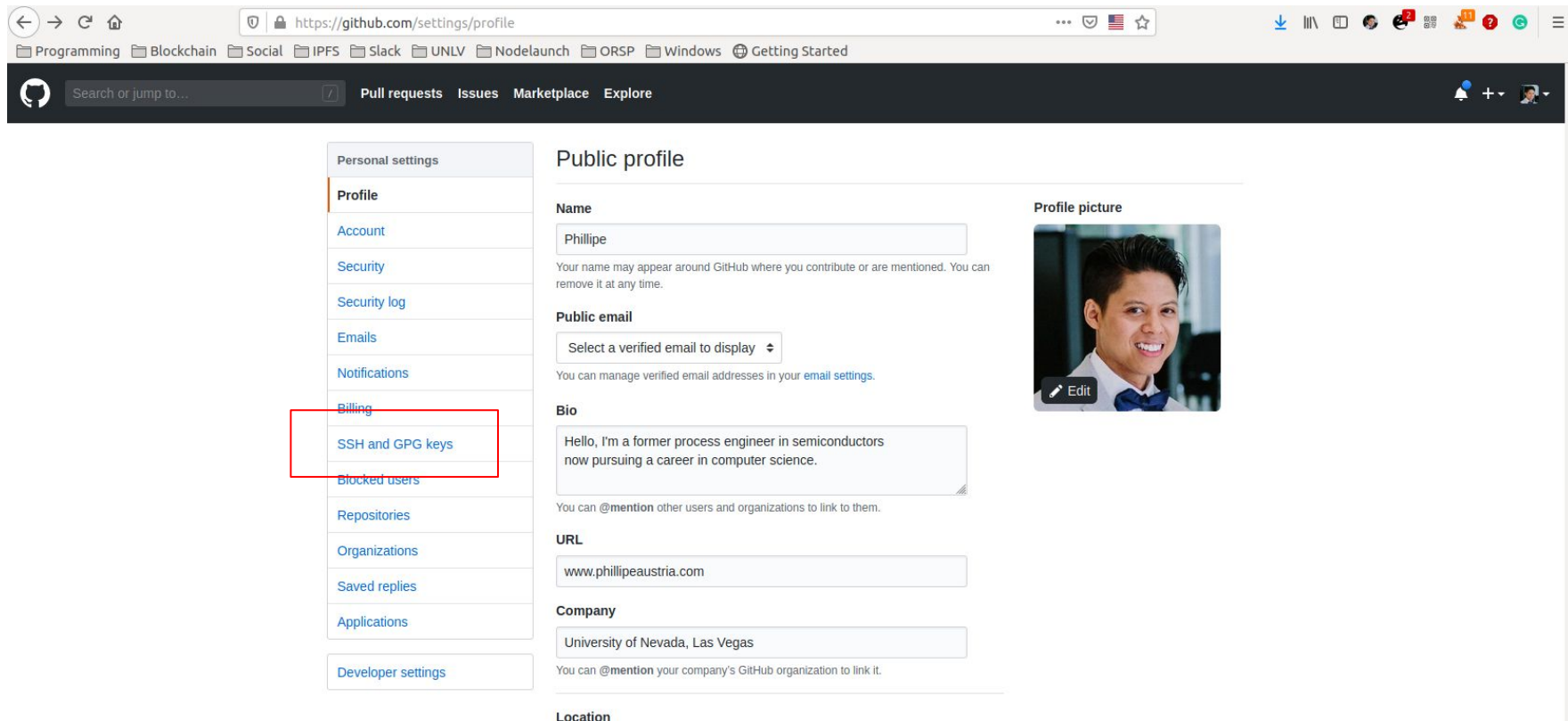
```
~ $ cat id_rsa.pub
```

Answer the following questions

```
Enter file in which to save the key  
(/home/demo/.ssh/id_rsa)
```

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

Placing it in your Github



The screenshot shows the GitHub settings page for a user named Phillippe. The left sidebar contains a list of settings categories: Personal settings, Profile, Account, Security, Security log, Emails, Notifications, Billing, SSH and GPG keys (highlighted with a red box), Blocked users, Repositories, Organizations, Saved replies, Applications, and Developer settings. The main content area is titled 'Public profile' and contains several sections: Name (Phillipe), Public email (Select a verified email to display), Bio (Hello, I'm a former process engineer in semiconductors now pursuing a career in computer science.), URL (www.phillipeaustria.com), Company (University of Nevada, Las Vegas), and Location. A profile picture of Phillippe is also shown.

Personal settings

- Profile
- Account
- Security
- Security log
- Emails
- Notifications
- Billing
- SSH and GPG keys**
- Blocked users
- Repositories
- Organizations
- Saved replies
- Applications
- Developer settings

Public profile

Name

Phillipe

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

Select a verified email to display

You can manage verified email addresses in your [email settings](#).

Bio

Hello, I'm a former process engineer in semiconductors now pursuing a career in computer science.

You can [@mention](#) other users and organizations to link to them.

URL

www.phillipeaustria.com

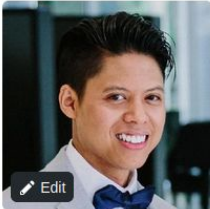
Company

University of Nevada, Las Vegas

You can [@mention](#) your company's GitHub organization to link it.

Location

Profile picture



Edit

Placing it in your Github

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



id_rsa.pub

da:c6:57:9f:99:50:6a:cd:73:52:cf:4c:53:0b:c7:f7

SSH

Added on Apr 9, 2018

Last used within the last 4 months — Read/write

Delete



skloolsh-nuc

be:c5:f2:0a:a2:f2:4a:9a:ea:7d:04:0b:75:8f:23:15

Added on Jul 16, 2019

Delete

Pull requests Issues Marketplace Explore

Personal settings

Profile

Account

Security

Security log

Emails

Notifications

Billing

SSH and GPG keys

Blocked users

Repositories

Organizations

SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Add SSH key