UCLA CS35L

Week 2

Wednesday

Reminders

- We are going over HW2 in class this week, and it is due next Monday (April 13)
- The HW Page for Assignment 2 was updated last week, make sure you see the current version. If it says you need to submit 4 files at the bottom, then it is correct.

- Anonymous feedback for Daniel - https://forms.gle/tZwuMbALe825DBVn8

SSH Cont

SSH Reminders

SSH Multi-Hop requires ssh-agent to be running

- You will need to start ssh-agent again and add your key at the start of your session

SSH X-Forwarding

- Make sure you have installed the X-Server on your local machine
- Make sure you enable X-Forwarding in your SSH session either with :
  - MAC: -X terminal flag
  - Windows: Putty checkbox

SSH Tips

- Typically we use a config file in the .ssh directory. The config file can specify:
  - Which key/username to use for which server
  - Any proxy commands (for multi-hop)
  - Any Keep-Alive settings (to stop your session from timing-out)
  - Port information
  - and more

- SSH Keys can be created without passwords, which makes them easier to use. But less secure.

A few more things about SSH

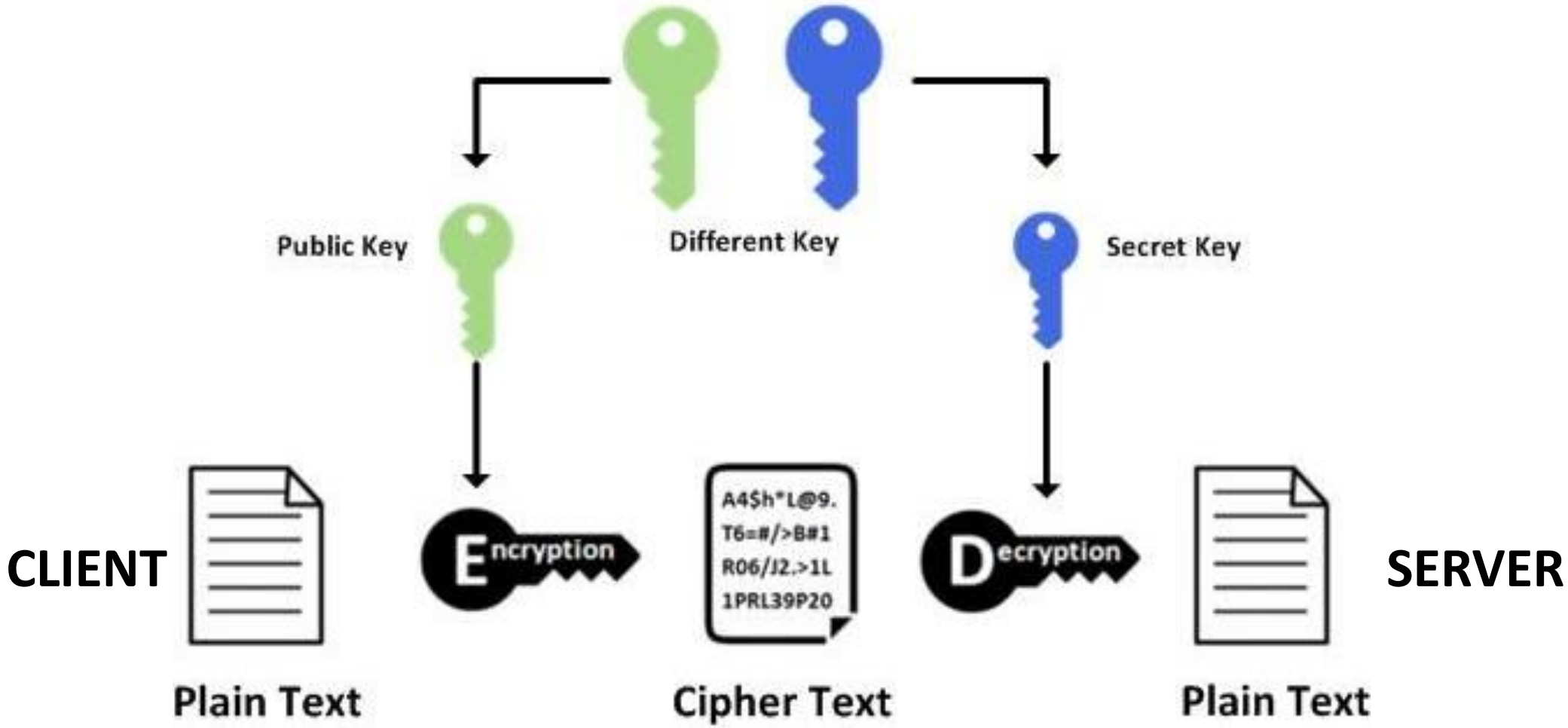
- For Windows Users
  - Try ditching Putty for WSL (Windows Subsytem for Linux)
  - Lets you get access to a Linux Kernel on your own machine for access to linux development tools, and things like OpenSSH from this lab
- For Mac/Linux/WSL Users
  - Check out keychain, which can automate even more of your ssh-agent workflow

GPG - GnuPG

Review of Encryption

- Symmetric
  - There is a shared secret
  - The same key is used to encrypt and decrypt data
- Asymmetric
  - 2 different (but closely related) keys – Public and Private
    - Only the creator knows the relation. The Private Key cannot be derived from the Public Key
  - The Public Key is shared with everyone
  - The Private Key is only stored locally, and never shared.
  - The Public Key encrypts data that the Private Key can decrypt and…
  - The Private Key encrypts data that the Public Key can decrypt

Asymmetric Encryption



What is GPG

- Stands for GnuPG, is the free and open-source version of PGP
  - PGP created in the early 90s and stands for “Pretty Good Privacy”
- Typically used for encrypting and verification of one-time data:
  - Text
  - Email
  - Files
- In contrast to SSH or SSL which establishes and sets up a secure connection to exchange encrypted data until the connection closes

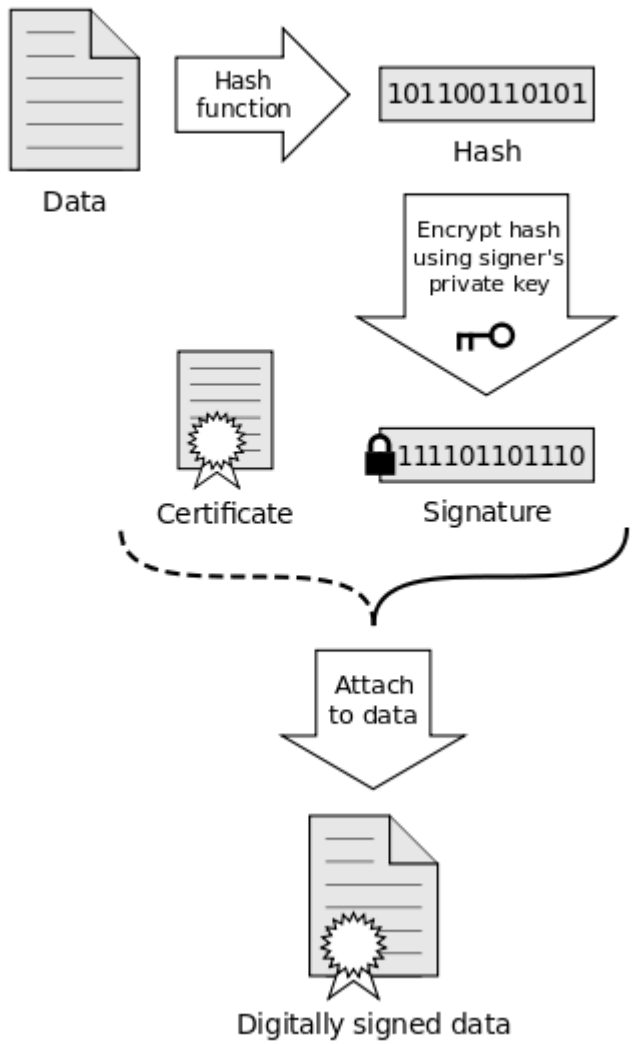
How does GPG work

- Uses similar Private and Public Key log to encrypt, sign, and verify files
- Keys are created and stored in the GPG Keyring
  - Located in ~/.gnupg
- Many different command options to use GPG:
  - Generate keys (stored in keyring)
  - Import file to keyring
  - Export key to file
  - Sign and verify data

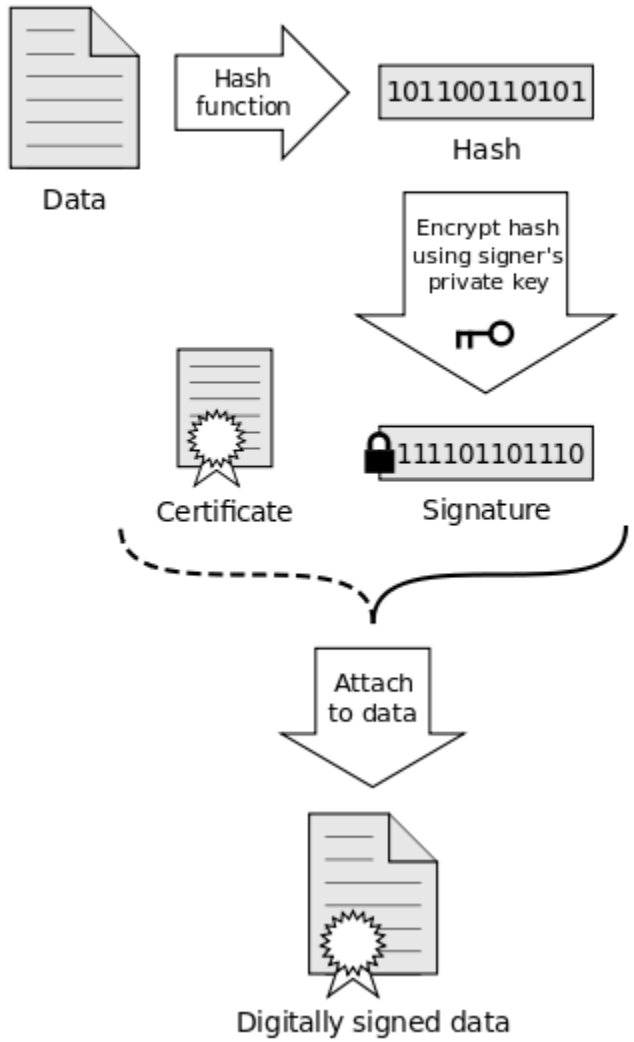
Signatures

- We covered encrypting/decrypting messages on Tuesday.
- But we also want to distribute files, and verify they came from the right person and have not been modified.
- It’s like “signing” a file

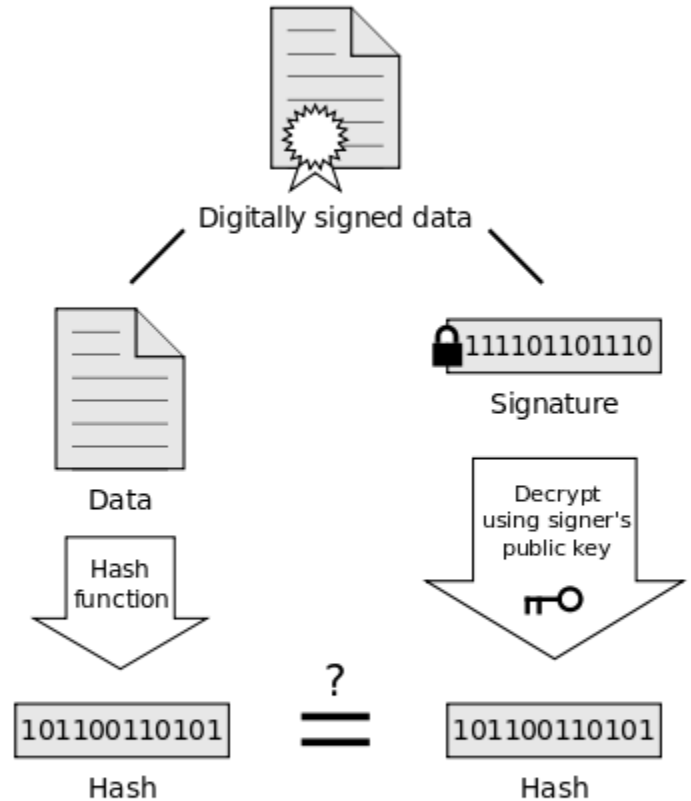
Signing


Hash function
101100110101
Hash
Data
Encrypt hash using signer's private key
111101101110
Signature
Certificate
Attach to data
Digitally signed data

Signing



Verification



If the hashes are equal, the signature is valid.

Generalized Encryption Functions

- At a high-level then, for secure communication we can imagine the following 4 functions:
  - Encrypt(plainText, publicKey) → cipherText
  - Decrypt(cipherText, privateKey) → plainText
  - Signature(data, privateKey) → signature
  - Verify(data, signature, publicKey) → True/False

Homework 9 - GPG

Log on to a SEASNet Server

- There is no client and server like in the lab – so log on to any lnxsrv0[6-9] or lnxsrv10 of your choosing.

Create a Key with GPG

- We are using GPG v2
- What is the generic command to generate a key?

Export the Key into a file

- By default, keys created go into the GPG-Keyring for security
- To share your public key though, you need to export it to a file
- What is the export command?

Generate a Detached Signature

- What is a detached signature?
- What key (Public or Private) is used to generate a signature?
- What is the command to generate the detached signature?

How to verify Detached Signature

- What key (Public or Private) is required to verify a signature?
- What GPG command is used to verify a detached signature?

- Does the gpg verification command work if you run it on another machine that does not have your public key?

Homework Question 1

- Question's 1 focus is on the SSH part of the Lab. Try to answer the following:

1. Now that you have SSH Keys and SSH-Agent setup, can a hacker sniff your network traffic to gain access to your account or to the server
2. If the hacker now installed a keylogger, can they get access to your account/server now
3. If you stored all of your client keys on a USB drive, and the hacker gets that USB drive. Can they access your account/server now.

NOTE – Cybersecurity is *usually* not a Yes/No world. What we are trying to figure out is if we are at a **higher** or **lower** risk because of our actions.

Homework Question 2

- Question's 2 Focus is on the weakness of gpg --verify, and the general verification/signature process.
- Try to examine all the steps involved, are there any weak links?
  1. The sender generates data
  2. The sender generates a signature for the data using its own private key
  3. The sender shares the data + signature + public key with the receiver
  4. The receiver verifies using the data + signature + public key

Questions