SEC 150 Assignment: Privacy and Authentication

Titus Barik (tbarik@staff.waynecc.edu)

Version 1.0

1 Assignment Description

In this assignment, you will work with public-key cryptography, and use GNU Privacy Guard (`gnupg`) to sign, encrypt, verify, and decrypt files. GNU Privacy Guard is largely compatible with PGP. PGP is commonly used with e-mail, but is general-purpose enough to be used for traditional encryption and signing tasks.

2 Pre-Requisites

Before starting this assignment, watch the lecture video on GNU Privacy Guard. You will use the `gpg` application provided by DSL; this must be installed before starting the assignment. The package is named `gnupg-1.4.7.dsl`.

3 Instructions

1. Generate a public and private key pair (DSL and Elgamal) that does not expire with your name and Wayne Community e-mail address using `gpg --gen-key`. Use a key size of 2048 bits, and whatever password you want (per GPG protocol, I never need your private key). Then, use `gpg --list-keys` and `gpg --list-secret-keys` to verify that your key information is correct.

2. Take the file `leaves.txt` and encrypt it with your newly generated key (using `--armor`). The file should be called `leaves.txt.asc`. Verify that the file `leaves.txt.asc` contains encrypted data.

3. Take the file `frost.txt` and sign it using the option `--clearsign`. Note that you are asked for your password, unlike before. The file should be called `frost.txt.asc`. Verify that the file is signed, but not encrypted. Signing and encryption serve different purposes. Signing proves that you are the person who you claim you are; encryption is used to hide data from prying eyes.

4. If you like, you can try to decrypt your encrypted file (of course, you already know what the contents are in this case, since you are the one who just encrypted it!).

5. Now, you will work with another user's public key. Import the SEC150 public key, called `sec150_class.pubkey`.

6. Verify that the files `raven1.txt` and `raven2.txt` are actually from SEC 150. One of these files has been tampered with; delete it, since it can't be trusted.

7. Take the correctly signed file, and open it. Strip the GPG information, so that all you have is the text. Now, add your name to this file. Then, encrypt this file using the **SEC150** key (not your own key!), and call it `raven.asc`. Observe that you are able to encrypt files for keys that don't belong to you; this is the beauty of public-key encryption. You can encrypt a file that you yourself can longer decrypt, and thus this is truly a secure communication! In this case, only SEC 150 (that is, me) will be able to decrypt your file, since only I have the private key for SEC150.

8. Export your public key (**never your private key**, which should not be shared with anyone) using `gpg --export --armor`. Be sure to specify your last name as the final argument, otherwise ALL keys will be exported, which is not what you want. Save this output as `lastname.pubkey`, where last name is of course your last name.

4 Milestone

Submit a screen capture showing that GPG has been installed on your local machine.

5 Submission

Submit the following files as a single zip file: `leaves.txt.asc`, `frost.txt.asc`, `raven.asc`, and `lastname.pubkey`.