Computer Science 530 - Lab Assignment #1 -- Fall 2021

Due: Friday September 17, 2021, 4:30 p.m.

In this lab you will use the gnugpg public key utility program to create and distrbute keys for users and interact with the keys and messages cryptographically.

In so doing, you will be using an application to utilize public key cryptography, as we discussed in the second lectue of CSci530. You will also be performing some aspects of key management, a topic that is covered in Lecture 3 of CSci530.

While last years version of this lab used VirtualBox Virtual Appliances to run these programs, it is alos possible to install gnugpg directly on your personal computer. Given the significant differences in the ability of some students to install VirtualBox because of new hardware architectures, I have recast this first exercise with instructions that will enable you to perform the lab through the GPG application which you will install directly on your computer.

For future labs, we will still be using the virtualization environment, after I have made alternate arrangements for students with M1 MACS to access the environment in other ways.

1. Setting up your computer to use GPG

   You can find some basic information about GPG from: https://gnupg.org/.

   You will want to visit the Download page at https://gnupg.org/download/index.html

   Where you will install the latest release and associated libraries for your hardware archiecture and operating system. Note that the section of this page labeled GnuPG BINARY RELEASE contains links to common distubutions for various platforms. While this is not the prefered way to install such software, it is the way that most users install the packages in the fewest steps. Note that the current version of GPG is 2.3.2.

2. Enter the Shell of your Operating system:

   If on windows, this is the "Command Prompt". If linux based, including MacOS, this will be a shell

3. Make sure GPG is in your Search Path
4. Type gpg -help to see a list of options
5. You will now generate a key pair for your use in this exercise.

   Technically, what you are doing is generating a key pair for a public key cryptosystem. The key pair will consist of both a private key, which you will retain for your own use, and a public key which will eventually be sent to a public key repository.

   Type "gpg --gen-key"

   You will be prompted as follows:

   - Real Name: Enter Your name in the form that you want it to appear for this assignment
   - Email address: Enter your USC email address
   - Enter "O" or Okay (correct)
6. You will be promoted to enter a passphrase

   You will also want to generate several movements with your mouse, based on which the GPG program will gather "random" data to use in key generation. GPG will generate a key pair, and it will store the private key from the key pair separately encrypted using the passphrase you enetered, so that someone finding your laptop is not able to easily determine your private key. You will be asked for this passphrase in the future when you attempt to perform an operation that requires your private key.

7. Verify that GPG has stored your key using the command:

   gpg --list-keys [optional: your email address]

   If you have used GPG previously you may have other keys stored locally. You can add an argument with your email address at the end of the gpg --list-keys command to list only keys associated with your email address. Verify that a key has been generated for your name, and the key ID and expiration associated with the key.

   Answer 1: Please save the output of the list keys operation (with your email address) above and submit as the first item in the solutions that you will upload.

8. In order for your public key to be used by others (to send you an encrypted file, or to validate your digital signature), it is necessary to upload your public key to a public key store.

   To do this you must first convert your public key to a format that may be distributed to other uers, or uploaded to a keystore.

   You do this with the command:

   gpg --export -a -o keyfile.txt [your email addres]

   which creates keyfile.txt containing a base64 encoded block of text embedding your public key.

   You may upload this to the open PGP keystore by visiting the website:

   https://keys.openpgp.org

   Once you upload the key file, to make it discoverable using your email address, you must click the button to verify your email address, and a message will be sent to you through which you an approve the publication of the key as associate with your address. Note that this is not strong authentication of the identity of the submitter of a key, it simply enables search. For you to reliability assess the integrity of the key itself you should exchange the "fingerprint" of your key through external means with those that might use the key to send you messages. We are not taking this final step in this lab exercise.

9. To encrypt a message that is to be viewed by another user you must obtain their public key. You will use the keyserver above, i.e.:

   https://keys.openpgp.org

   to retrieve the public key for csci530@usc.edu.

   ANSWER 2: Please record the full key fingerprint for this file (it is the last part of the link that is returned in the search you just performed), and include this as answer 2 of your lab submission.

   Also, verify that the fingerprint is accurate by making sure that the last 6 characters are D83E47. You typically should verify the full fingerprint with the intended receiver out of band (i.e. in person, by phone, or through other means) to make sure that an imposter had not uploaded their own key to the keyserver in an attempt to impersonate the subject named in the key.

   Load this key into your local keystor with the GPG command:

   gpg --import [filename where you downloaded the key]

10. Now that you have all of the necessary keys in place let's use them for their intended purposes. GPG's intended purposes are two: encrypting (for confidentiality) and signing (for authentication and data integrity).

    Encrypting and decrypting

A user encrypts for consumption by another user, by applying the other user's public key to the plaintext. The result is decryptable by that particular other user only.

So, you should create a file with a name of your choosing and with the content:

> My email address is [your email address] and my key fingerprint is [your key fingerprint]. I am sending this message to csci530@usc.edu

You can create an encrypted version of this message that will be readable only by csci530@usc.edu with the command:

gpg --recipient csci530@usc.edu -a --encrypt [the filename you used to create the file above]

(the -a option tells it to output in BASE64 encoding rather than binary)

Note the creation of the file [filename].asc which contains the encrypted data.

ANSWER3: Add the contents of .asc to your submission file as the answer to item 3.

11. You will also want to digitally sign your message, so that I can be certain that it comes from you. You may use the --clearsign option to apply only a digital signature, but in this example, we are going to complete both the encryption and the digital signature in a single command. Note also, that in this example, what GPG is doing is calculating a hash over the message and encrypting the hash using your private key, rather than directly encrypting the entire message.

    The command you will use is:

    gpg --sign --recipient csci530@usc.edu -a --encrypt [filname]

    Which will create [filename].asc

    Containing all of the elements necessary for both confidentialy and the digital signature (integrity).

    Please compose an email to csci530@usc.edu and include the [filename].asc in the body of your message.

    I would prefer if you cut and paste the text from the .asc file into the body of your message, rather than have you attach it as a separate attachement, since this will make it easier for me to process the message within my mail reading program..

12. I will reply with a digitally signed and enrypted message which you can then store in the file:

    encsignedresponse.asc

    You will verify my signature using the csci530@usc.edu key and decrypt the message using your private key using GPG commands --decrypt and --verify (I will leave it to you to figure out the correct use if these options).

    ANSWER4: Please include the text of my response and the output valdating the signature in the fourth submitted item for you lab.

INSTRUCTION:

The report must be submitted throuh the D2L Dropbox Folder for Lab Assignment 1.

How to submit Assignment #1:

- STEP 0. Be sure to include you name in the body of your assignment.
- STEP 1. Please login to DEN, and select csci530.
- STEP 2. Please select "Assignment" in the menu.

- STEP 3. Please select "view/Complete Assignment #1".
- STEP 4. Please select "File To Attach" to attach your report. (NOTE: PDF, MS WORD, ASCII TEXT ONLY! Other formats are NOT acceptable.)
- STEP 5. Please select "Submit" button. (If you select *SAVE* button instead of *submit*, then the TA cannot view your report for grading.)