

Jackson State University
Department of Computer Science
CSC 437-01/539-01 Computer Security
Fall 2013

Instructor: Dr. Natarajan Meghanathan

**Lab Project # 1: Lab Project on using PGP – GNU Privacy Guard (GPG) for
Secure E-mail Communication**

Due: October 8, 2013: 7.30 PM

Pretty Good Privacy (PGP)

PGP is a secure E-mail communication standard that provides cryptographic privacy and authentication. PGP supports both symmetric keys-based and public/private key pairs-based secure communication between communicating parties. In this project, we will do both symmetric as well as public key-based encryption. We will use GNU Privacy Guard (GPG) – an open-source version of PGP in our project.

WHAT TO SUBMIT: Read through the entire project description and follow the steps as indicated in detail. Submit the following

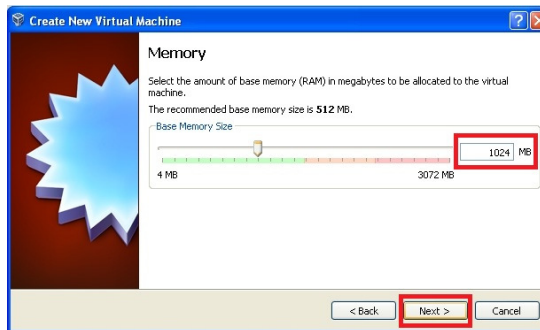
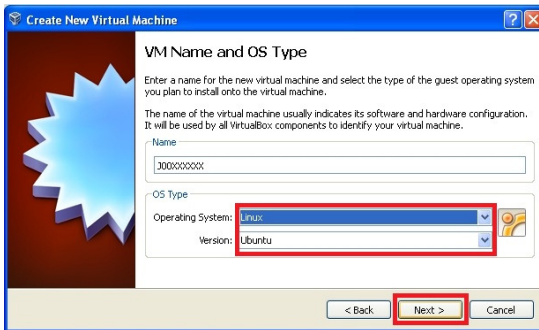
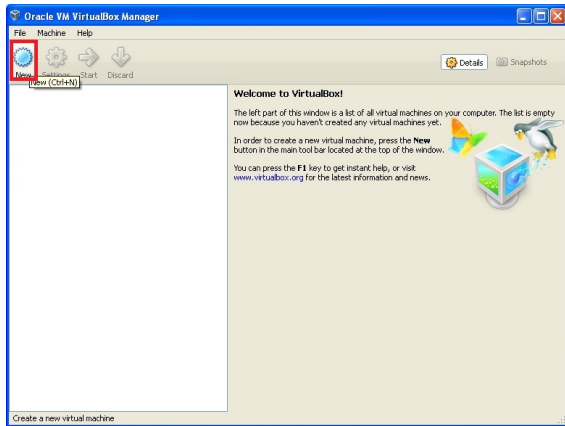
1. **An abstract** (8 points) – briefly describe the actual project, in about 200-250 words
2. **Introduction** (12 points, about 0.75-1 page): Compare the two e-mail security standards: PGP and S/MIME, including their working principle, applications, pros and cons, etc.
3. **PGP in detail** (20 points, about 1.5 pages): Explain the working of the PGP for both symmetric and public-key encryptions (using a flowchart followed by description), for each case, indicating the sequence of steps, in a nutshell.
4. **Answer the questions** (20 points) 1 through 6 as they appear in the course project description.
5. **Project Execution Screenshots** (40 points): screenshots of all the numbered figures, starting from page 5 (1 through 22, for sure, and 23 through 36, depending on how much you can proceed with regards to question 6). Submit your detailed project report with Figures 1 through 36 **clearly labeled**. When you take the screenshots, make sure you place your mouse pointer outside the Virtualbox window (somewhere on your local computer) with the Virtualbox window (Ubuntu screen) still open to capture what you want.

You will do this project in a virtual machine environment.

Installing VirtualBox 4.2 and Ubuntu OS

Go to <https://www.virtualbox.org/wiki/Downloads> and download VirtualBox for your operating system. If you work on a lab computer, you need to use the Ubuntu VM .iso file that is stored on the local machine. If you work on your personal computer, you need to download the Ubuntu .iso file from the website listed in Step # 1 and continue. You may use the following steps for installing the Ubuntu VM on the virtualbox.

1. The Ubuntu installation file is located on the desktop of your PC in the lab machines (it can be downloaded from <http://www.ubuntu.com/download/ubuntu/download> if the .iso file cannot be located on your desktop).
2. On the VirtualBox Manager screen click on “New”



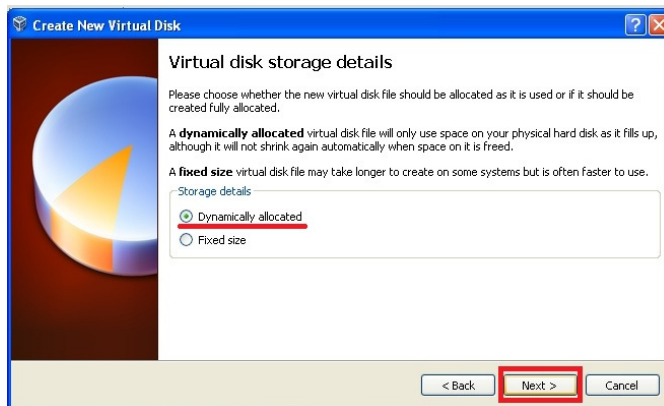
3. When prompted, put your J # for the name of the VM and select “Linux” as OS (when you choose Linux as OS, the program should automatically choose Ubuntu as Version, if not select Ubuntu) and click Next.
4. Set the RAM memory at 1024 MB on the following screen and click Next.
5. On the following screen make sure that “Start-up disk” is selected. Check “Create new hard disk” if not selected.



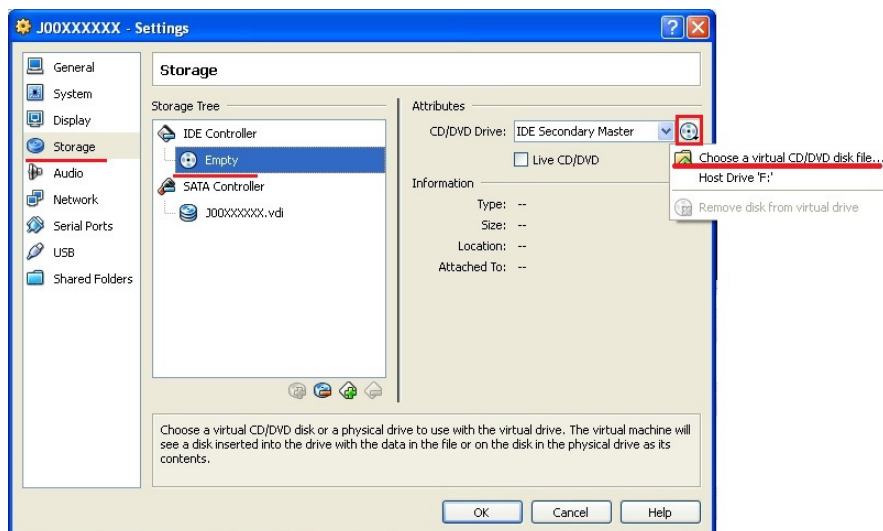
6. On the following screen select “VDI (VirtualBox Disk Image)” and click Next



7. On the next screen, make sure to select “Dynamically Allocated” and click Next (this option will enlarge the hard disk space as needed).



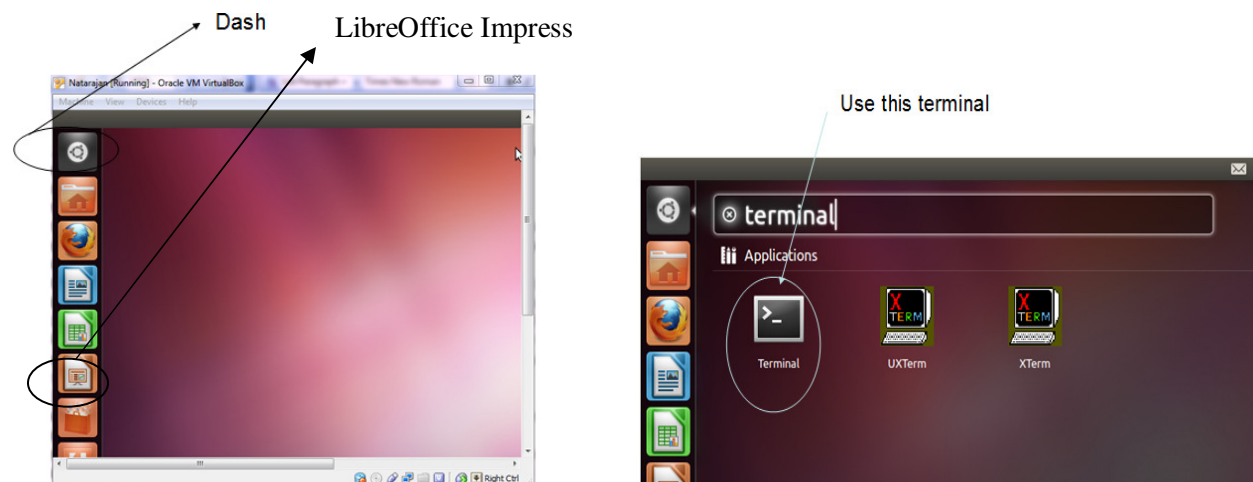
8. Leave everything as it is on the next page and click Next.
9. The next screen displays the information about the VM you are about to create; click “Create” twice and wait for VirtualBox to create your VM.
10. After your VM appears on VirtualBox Manager’s screen highlight the VM and click on “Settings.”
11. On the popped-up screen select “Storage.” In the Storage menu select “Empty” on Storage Tree under IDE Controller. When “Empty” is selected, click the circular disk object under Attributes and click on “Choose a virtual CD/DVD disk file.”



12. On the popped-up window show the location of the Ubuntu image file and click OK (by default the Ubuntu is on your desktop, if you have downloaded it from the website, show the path to the downloaded file accordingly).
13. The “Empty” under IDE Controller should have the name of the Ubuntu file, if you have done everything listed above. If it does click OK, otherwise redo steps 2 to 13.
14. Now, it is time to install the Linux OS on the VM. To be able to do that, highlight the VM and click on “Start” or simply double click the VM.
15. Click OK, on series of tips that pop-up. Wait till the Welcome window appears inside the VM (ignore the message in red before the Welcome page).
16. On the Welcome page select “Install Ubuntu”
17. On the next screen click “Forward”
18. On the next screen select “Erase disk and install Ubuntu” and click Forward (do not panic about the warning, the disk mentioned is allocated for VM, so no actual hard drive is erased).
19. Click “Install Now”
20. While installation process is going, you will be prompted to choose your time-zone. Just type the name of the city, and it should give you the choices from which you can select the appropriate one, and click Forward; on the keyboard layout leave everything as it is, and click Forward.
21. On the “Who are you?” page, type your initials and last 4 digits of your Student ID (J#) for “Your Name” box. The system should automatically fill your computer name and a user name. For your password, type a sequence of characters of your choice and click “Forward.”
22. Wait until the virtual OS is installed, it may take some time, be patient.
23. After installation is complete, you will be prompted to restart your system. Click on Restart Now and then press any key as prompted (it only restarts VM)
24. In Login page click on your username (your username should be your initials and last 4 digits of your J#) and type your password.
25. Ignore the pop-up and click Close.

Starting the Ubuntu Virtual Machine

Now, double click on the VM on the left side screen that you have got and keep clicking OK for all the messages that come (you can even make them not to show up from next time, if you wish so) and now the Ubuntu VM would have been loaded up (left screenshot below). Now open a terminal window by selecting load up the terminal window, by clicking the Dash icon (the top most icon in the window below, as indicated) and type ‘terminal’ on the search space that pops up. Click on the left most terminal icon (the pure black one) that shows up and launch the terminal to start working on your project.



Stage 1: Installing GPG on your Ubuntu Virtual Machine

1. In the terminal, run the command “`sudo apt-get install pgpgpg`”. When asked, enter the password that you used during the Ubuntu installation. The GPG installation will start and wait for it to complete.
2. Following step 1, run the command “`sudo apt-get install gnupg-agent`” in the terminal. The GnuPG-Agent installation will start and wait for it to complete.

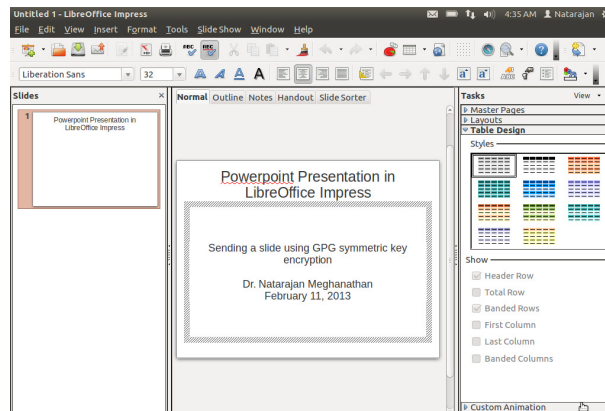


Figure 1: Powerpoint Presentation in LibreOffice Impress

Stage 2: Symmetric Key Encryption/ Decryption with GPG

In this phase of the project, we will create a Powerpoint (.ppt) file using the *LibreOffice Impress* application available in Ubuntu. We are going to encrypt the .ppt file at one end using GPG and a symmetric key and send it as a secure attachment over e-mail to the other end where we will decrypt using GPG and the same symmetric key, extract and view the .ppt file.

1. From the menu on the left side panel, open the LibreOffice Impress Powerpoint application.
2. Type the contents in a slide, something similar to the one shown in Figure 1, with your name and date appearing on it. Save the file as a .ppt file. Note down the location where you save the file. In this project description, I am storing it in the folder `/home/natarajan/gpg`. You can cross check by going to the terminal and to the `/home/natarajan/gpg` folder and make sure the .ppt file is there. I have named the .ppt file as `NM-file1.ppt`, where ‘N’ and ‘M’ are the first characters of my first name and last name respectively. You should also follow a similar convention, based on your first name and last name, while doing your project.

Note: By default, you would be logged into the `/home/natarajan/` folder. You may want to create a ‘gpg’ folder for yourself to do this project (you could create a gpg folder using the `mkdir` command in the terminal or in the file saving window), though this is not mandatory. You can save your file anywhere inside any folder.

3. Now, run the command “`gpg --symmetric NM-file1.ppt`”. You will be prompted for a passphrase and to repeat what you entered again. In this project description, I used `secprj` as the passphrase. You are free to use anything that you wish. But, remember that passphrase as you will need it to decrypt. Now run the `ls -l` command on your terminal, you should be able to see a .gpg file. In Figure 2, it appears as `NM-file1.ppt.gpg`.

```
natarajan@natarajan-VirtualBox:~/gpg$ ls -l
total 80
-rw-rw-r-- 1 natarajan natarajan 72704 2013-02-11 04:39 NM-file1.ppt
-rw-rw-r-- 1 natarajan natarajan 5992 2013-02-11 04:51 NM-file1.ppt.gpg
natarajan@natarajan-VirtualBox:~/gpg$
```

Figure 2: GPG Symmetric Key Encryption of the Powerpoint File

```
natarajan@natarajan-VirtualBox:~/gpg$ cp NM-file1.ppt.gpg email-NM-file1.ppt.gpg
natarajan@natarajan-VirtualBox:~/gpg$ ls -l
total 88
-rw-rw-r-- 1 natarajan natarajan 5992 2013-02-11 05:13 email-NM-file1.ppt.gpg
-rw-rw-r-- 1 natarajan natarajan 72704 2013-02-11 04:39 NM-file1.ppt
-rw-rw-r-- 1 natarajan natarajan 5992 2013-02-11 04:51 NM-file1.ppt.gpg
natarajan@natarajan-VirtualBox:~/gpg$
```

Figure 3: Copying the encrypted GPG file and saving as another file

4. You can actually e-mail this NM-file1.ppt.gpg file as an attachment and it can be downloaded and saved at the receiver side in a different file name, but of course with the “.ppt.gpg” extension. In this project description, I have just copied the NM-file1.ppt.gpg file to another file with name email-NM-file1.ppt.gpg in the same folder. You can use the “cp” command as I have shown. Use ls –l to see if the new file is there. See Figure 3 above.
5. Run the command “gpg --output extracted-NM-file1.ppt –d email-NM-file1.ppt.gpg” as shown in Figure 4, you will be prompted for the passphrase. The extracted contents will be in the extracted-NM-file1.ppt file. You can cross-check by opening the extracted .ppt file.

```
natarajan@natarajan-VirtualBox:~/gpg$ gpg --output extracted-NM-file1.ppt -d email-NM-file1.ppt.gpg
gpg: keyring `/home/natarajan/.gnupg/secring.gpg' created
gpg: CAST5 encrypted data
gpg: encrypted with 1 passphrase
gpg: WARNING: message was not integrity protected
natarajan@natarajan-VirtualBox:~/gpg$ ls
email-NM-file1.ppt.gpg extracted-NM-file1.ppt NM-file1.ppt NM-file1.ppt.gpg
natarajan@natarajan-VirtualBox:~/gpg$
```

Figure 4: Decrypting and extracting the Powerpoint file

```
natarajan@natarajan-VirtualBox:~/gpg$ gpg --symmetric --armor NM-file1.ppt
```

Figure 5: Encrypting the Powerpoint to printable ASCII characters

```
natarajan@natarajan-VirtualBox:~/gpg$ gpg --symmetric --armor NM-file1.ppt
natarajan@natarajan-VirtualBox:~/gpg$ ls -l
total 172
-rw-rw-r-- 1 natarajan natarajan 5992 2013-02-11 05:13 email-NM-file1.ppt.gpg
-rw-rw-r-- 1 natarajan natarajan 72704 2013-02-11 05:14 extracted-NM-file1.ppt
-rw-rw-r-- 1 natarajan natarajan 72704 2013-02-11 04:39 NM-file1.ppt
-rw-rw-r-- 1 natarajan natarajan 8213 2013-02-11 05:16 NM-file1.ppt.asc
-rw-rw-r-- 1 natarajan natarajan 5992 2013-02-11 04:51 NM-file1.ppt.gpg
natarajan@natarajan-VirtualBox:~/gpg$
```

Note the difference in the file sizes. Why is it?

Figure 6: Encrypting the Powerpoint to printable ASCII characters

Question 1: Why is there a difference in the file sizes of the .asc and .pgp versions in the symmetric key encryption?

6. Now, instead of attaching the encrypted .pgp file, if you want to send the encrypted data as part of the e-mail body, then you can encrypt the NM-file1.ppt using the command: “gpg --symmetric --armor NM-file1.ppt”. See Figure 5. You will be prompted to enter a passphrase and repeat it. You can enter a different passphrase or the same passphrase that you used in Step 3. But, remember the passphrase you entered now. An encrypted file by the name NM-file1.ppt.asc would have been created.
7. A run of the ls -l command will list the NM-file1.ppt.asc (see Figure 6) and if you run the cat command as: “cat NM-file1.ppt.asc”, a long list of printable ASCII characters would be displayed (see Figure 7). You can open your email compose box and copy and paste the contents of the NM-file1.ppt.asc and send it to the receiver. Only with gpg and the correct passphrase, someone can decrypt the message.

```
natarajan@natarajan-VirtualBox:~/gpg$ cat NM-file1.ppt.asc
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.11 (GNU/Linux)

jA0EAWmC8t7wYctG5+Bgyexa04BSRy3V3AEAQJJ2PkTGrL+7gbk1Hb0cD6RMMkpp
td+ot+YhmeGgBnfq0sQhUTUfbPczvpU0Ki2j3f9YFJamclgVThdK4wsNUzynwV2v
x/VRIZy9FydiFE8AxJ9kUAcGt+Aah3DmbQYWNidUPTKth59UR2tRrFgiMFYQC/Tm
o/hI8I1A2E2Luer3E6RYtNw9JVAZgBo5xsIJdvUtArV8Zuju4bmmkHFA66kmCk04
bhfLWbfjDh2A5DgxpMPbnsukB2nA+gDilja00wgjKfHFRZitaJatAgsc0B7GeYMT
Hc9RBYJrILEbQsScoo/QBt354gJ7SCV+6DMuF25yozra86w3In0jAtNy0B6jF9dB
ldPkU8tuFnIjJb+4jM2Ym3qL91fzunEx1ed77JwxKuv2b4NcgUc2nYWSiAk2Pxt4
dPt0vzf14jQjFKNSjeYNUiYJrpYULCTXLaeWo7BTXdoVZE+tIwrwpHdwGjwtrqFd
akmu0+Pcdoq/RgrX7a/uQ/ZXoEPRGqyjiEdGrALYhXocnIY+59mxQqq7RXyAnzhX
/B+fqAyNlTyELoAy5qmr/3Pj57rJxEHpFLJAnqUtpqwf28P9dmqXC8JRokj4/jcu
T/rpGT/yArQp6iA20BeACU20DbR7EiliuB81KAmMkzG1RS/NPpgRBFc984VWJzy
jLpPSer0GcSvTwoIEXaUUmXsyY5DLBCqBt68knrs0bHHH0Jr7s28kuoaSVsLGecI
irVgoT1C+NEvZP0musu6mu0Wg0EZVyeQ0p840GhJHNBqfMvGLJAjRPioIcdQZLyj
6ojbgv8A408r/PH0bCBvM9xvy5dujy13KwkyNchTccURLRIYw4AnnPI9pbx4kDBu
d6XtdNx+YN7T1GAhU0r7pTLDNdZruJULyZp+Lvu4QEfJJ7W1MaACfyvtFLjEZsZ
XN04VwDmAgw0I/fmRzkeJdkC3N6CFfy/Ejd/gQDzMc9qLlhHOV5FobuUaFSLVi
o8C+VoCT0BYWHofS7/PnsxVv/sVl4ogSBjQoWsXWJoGZZQDU7tpz2iHIUgpzpeXw
t0pZ2i71GYvTiTI1sQXm0qAaFvYPCqdTRLjKFVeIY4umvu/rREGJPIu+o2FxmEN
```

Figure 7: A portion of the encrypted PGP message of the Powerpoint file

8. To decrypt the encrypted file NM-file1.ppt.asc, run the gpg command as: “gpg --output ext-NM-file1.ppt --armor -d NM-file1.ppt.asc”. The extracted Powerpoint file would be stored with a file name ext-NM-file1.ppt. You can use the ls -l command to check whether the file has been extracted and is available in the folder. See Figure 8.

```
natarajan@natarajan-VirtualBox:~/gpg$ gpg --output ext-NM-file1.ppt --armor -d N
M-file1.ppt.asc
gpg: CAST5 encrypted data
gpg: encrypted with 1 passphrase
gpg: WARNING: message was not integrity protected
natarajan@natarajan-VirtualBox:~/gpg$ ls -l
total 244
-rw-rw-r-- 1 natarajan natarajan 5992 2013-02-11 05:13 email-NM-file1.ppt.gpg
-rw-rw-r-- 1 natarajan natarajan 72704 2013-02-11 05:25 ext-NM-file1.ppt
-rw-rw-r-- 1 natarajan natarajan 72704 2013-02-11 05:14 extracted-NM-file1.ppt
-rw-rw-r-- 1 natarajan natarajan 72704 2013-02-11 04:39 NM-file1.ppt
-rw-rw-r-- 1 natarajan natarajan 8213 2013-02-11 05:16 NM-file1.ppt.asc
-rw-rw-r-- 1 natarajan natarajan 5992 2013-02-11 04:51 NM-file1.ppt.gpg
natarajan@natarajan-VirtualBox:~/gpg$
```

Figure 8: Extraction of the plaintext Powerpoint file from the ciphertext ASCII character file

Stage 3: Public Key Encryption/ Decryption with GPG

Scenario: The scenario here is there are two users: *nmeghanathan* (referred to as the ‘sender’) and *natarajan_jsu* (referred to as the ‘receiver’), with e-mail accounts nmeghanathan@jsu.edu and mnatraj77@hotmail.com respectively. We are going to create new user accounts with usernames *nmeghanathan* and *natarajan_jsu* in our Ubuntu machine. We are then going to create the public-key and private-key pair for the two users. The receiver account exports its public key to the sender and the latter uses it to encrypt a secret message and send the encrypted version back to the receiver. Now, the receiver decrypts the ciphertext message with its private key and extracts the message. In this scenario, we are going to send a simple text file as our message. The details are explained below:

Accounts to be created by the students: Following the analogy described here, you should create two user accounts that somehow capture your first name and/or last name and that you are associated with JSU. Associate your JSU email address with the sender account and your non-JSU address (any active email address should work) with the receiver account.

1. In this projection description, I create two user accounts by name, *nmeghanathan* and *natarajan_jsu*.

Figure 9 illustrates the user account creation step.

Note: It does not matter from which folder you create the two user accounts. I just create it from the `/home/natarajan` folder. Note that you may also be prompted to enter the password (for Ubuntu OS) the first time you try to create an account. Then, for each of the two accounts, you will be asked to enter a UNIX password and reconfirm it. Choose a password of your choice for each of the two user accounts.

```
natarajan@natarajan-VirtualBox:~/gpg$ sudo adduser nmeghanathan
[sudo] password for natarajan:
Adding user `nmeghanathan' ...
Adding new group `nmeghanathan' (1001) ...
Adding new user `nmeghanathan' (1001) with group `nmeghanathan' ...
Creating home directory `/home/nmeghanathan' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for nmeghanathan
Enter the new value, or press ENTER for the default
  Full Name []: Natarajan Meghanathan
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
natarajan@natarajan-VirtualBox:~/gpg$
```

```
natarajan@natarajan-VirtualBox:~/gpg$ sudo adduser natarajan_jsu
Adding user `natarajan_jsu' ...
Adding new group `natarajan_jsu' (1002) ...
Adding new user `natarajan_jsu' (1002) with group `natarajan_jsu' ...
Creating home directory `/home/natarajan_jsu' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for natarajan_jsu
Enter the new value, or press ENTER for the default
  Full Name []: Natarajan M
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Is the information correct? [Y/n] y
natarajan@natarajan-VirtualBox:~/gpg$
```

Figure 9: Creation of the two user accounts *nmeghanathan* and *natarajan_jsu*

Question 2: What is the purpose of the sudo command?

Question 3: Where you prompted for the Ubuntu OS password when you tried to create the second account? If not, why is that you were asked for the first time and not the second time?

2. Run the command “sudo visudo” to launch the file “/etc/sudoers.tmp” in a text editor and insert the two lines as shown in Figure 10 (note that you will have to appropriately change the usernames depending on what you have created). Then, use Ctrl+O to save the file and press Ctrl+X to exit the editor. This step will make the two user accounts to be able to use “sudo”.

```
GNU nano 2.2.6 File: /etc/sudoers.tmp
# Host alias specification
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL
nmeghanathan ALL=(ALL) NOPASSWD: ALL
natarajan_jsu ALL=(ALL) NOPASSWD: ALL
```

Add these two lines

Figure 10: Editing the /etc/sudoers.tmp file to let the two new users to run the sudo command

Question 4: What do you think are the above two lines doing?

3. Login to nmeghanathan using the sudo command “sudo login nmeghanathan”. You would be probably prompted for the Ubuntu OS password and definitely for the password corresponding to the user account.
4. Launch another terminal and login to the other user account *natarajan_jsu*, in my case, through the command: “sudo login natarajan_jsu”
5. Generate the keys for the sender account: *nmeghanathan* using the command “gpg --gen-key”. Similarly for the receiver account: *natarajan_jsu*. Choose the values for the algorithm parameters as indicated below for the sender and receiver accounts. Make sure to associate an email address, when asked, to each of the two accounts.

Choose the following parameters for both the accounts:

Kind of Key: DSA and Elgamal
DSA Key size: 2048 bits
Keys does not expire at all

Then, for each of the two accounts, enter your real name and the email-address you want to associate with. Write a different comment for the two accounts.

4. Exporting the public key of the receiver (natarajan_jsu) to the sender (nmeghanathan): In this step, run the command: “gpg --armor --output mnatraj-pk --export mnatraj77@hotmail.com” to dump the receiver natarajan_jsu’s public key in file “mnatraj-pk” (see Figure 12). You can run “more mnatraj-pk” to review the ASCII formatted-version of the public key (Press ‘q’ to quit from the *more* screen). Now, run the copy command in sudo mode to copy the receiver’s public key mnatraj-pk to the sender nmeghanathan’s home folder. The command to be used is: “sudo cp mnatraj-pk /home/nmeghanathan” (see Figure 13). In the sender nmeghanathan’s terminal window, verify the existence of the file mnatraj-pk in the folder /home/nmeghanathan by running the ls command in that folder (see Figure 14).

```
natarajan_jsu@natarajan-VirtualBox:~$ gpg --armor --output mnatraj-pk --export m
natraj77@hotmail.com
natarajan_jsu@natarajan-VirtualBox:~$ more mnatraj-pk
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.4.11 (GNU/Linux)

mQMUBFEY7FgRCADkkAKZHtrl2V7BuZBTvsNJHDxwtehh2QiK/j9NIkB4otSsun6T
L3vquxmvyh8avVF3H8APXsBbh1cKWTxttaJhnr24JfsYmtE1aKK0dw0BGkLZ5sza
3FUZzSRY53xNWpITCjj7r6pBkqknJaqWPPHUR/8MvmnGThe09apIT8QirwLP/SSa
bK4zt6H7tSCDJhXD5E7G4w4Uqq0mMi04j1FTteGrENmssewnotEOqkC54J63tG+J
qyLm+BXiFa9b1fRgfdsi1REpSrLncPPoksFfqp/DyR9wgD36Wu0cehaJhs10Q0yz
uh6vUkNHqkwTjKs4nBA3SXlZ0YndIuyZ3RrvAQCW22j1002MeVIUYXVp/HQ88xmq
E+QJMG0y0mjd0TePtWqAo/8i+bywlvD+n0uDXsGCl19WJVJ72fyPzV/3WMAMLSdt
```

Figure 12: Exporting the Receiver (natarajan_jsu) public key to an ASCII file mnatraj-pk

```
natarajan_jsu@natarajan-VirtualBox:~$ sudo cp mnatraj-pk /home/nmeghanathan
```

Figure 13: Command to copy the Receiver’s public key to the Sender default folder

```
nmeghanathan@natarajan-VirtualBox:~$ ls -l
total 8
-rw-r--r-- 1 nmeghanathan nmeghanathan 179 2013-02-11 05:28 examples.desktop
-rw-r--r-- 1 root root 2288 2013-02-11 07:08 mnatraj-pk
nmeghanathan@natarajan-VirtualBox:~$
```

Figure 14: Proof of existence of the Receiver’s public key at the Sender’s default folder

6. Importing the Receiver’s public key to the Sender’s key store: Now that the receiver has exported its public-key file to the sender, the sender needs to import it. Run the command in the terminal of the sender nmeghanathan: “gpg --import mnatraj-pk”

```
nmeghanathan@natarajan-VirtualBox:~$ gpg --import mnatraj-pk
gpg: key 3AA67C15: public key "Natarajan M <mnatraj77@hotmail.com>" imported
gpg: Total number processed: 1
gpg: imported: 1
nmeghanathan@natarajan-VirtualBox:~$
```

Figure 15: Importing the Receiver (mnatraj-pk) public key to the sender (nmeghanathan) key store

7. Create and Encrypt a Message:

- Open pico or any text editor at the Sender (nmeghanathan) and enter the following line and save the file as secret-message.txt. Alternatively, you can also use the “cat > secret-message.txt” command and press Ctrl+D after entering the line (see Figure 16).

This file is created by Natarajan Meghanathan on February 11, 2013.

```
nmeghanathan@natarajan-VirtualBox:~$ cat > secret-message.txt
This file is created by Natarajan Meghanathan, February 11, 2013.
nmeghanathan@natarajan-VirtualBox:~$
```

Figure 16: Creation of the Text File to be sent

- Now, run the command (see Figure 17): “`gpg --recipient mnatraj77@hotmail.com --armor --encrypt secret-message.txt`” to encrypt the secret message text file using the public key of the user associated with the email address `mnatraj77@hotmail.com` (i.e., the receiver) and note that here we want to transform the encrypted version into an ASCII format. Press Y for accepting the public key of the receiver as stored in the key store. An encrypted version of the text file with name, `secret-message.txt.asc` would have been created. Use the `ls -l` command to see the presence of the encrypted file and run “`more secret-message-txt.asc`” to see the contents of the file (see Figure 18).

```
nmeghanathan@natarajan-VirtualBox:~$ gpg --recipient mnatraj77@hotmail.com --armor --encrypt secret-message.txt
gpg: 56A90FA2: There is no assurance this key belongs to the named user

pub 2048g/56A90FA2 2013-02-11 Natarajan M <mnatraj77@hotmail.com>
Primary key fingerprint: 4184 D28E C780 7B6F CD95 30C5 4B3F 5194 3AA6 7C15
Subkey fingerprint: 88EE 7187 C0DC BF40 A0CE 5D14 2F62 7B0F 56A9 0FA2

It is NOT certain that the key belongs to the person named
in the user ID. If you *really* know what you are doing,
you may answer the next question with yes.

Use this key anyway? (y/N) y
nmeghanathan@natarajan-VirtualBox:~$
```

Figure 17: Encryption of the secret message text file at the sender using the public key of the receiver

```
nmeghanathan@natarajan-VirtualBox:~$ more secret-message.txt.asc
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.11 (GNU/Linux)

hQIOAy9iew9WqQ+iEAgAsvyZ9JpFCnbCqhz8zXirSSTD+kK7NUqXtNG0Z/ZirEjb
Ib+6LCMT+eMNsKYG+NfkFn8n8gTsm3bfrB0zRfs2S98PoHynKNopDzyCRKHTZJI
rSjwQ0behMSvUoEhzck+PXJ5ed49jPCZ+jkmBTkuWslxHaTfiKqCsDx+/+2C481Kk
4MdaCBqFLv1H2CBG3myWN8qkThZqt0ioRCtczKIzqNYI/z8RC7YU4YcUa6L1Anye
npcZpE8P5dfw5zELV40K9oFdEW6VSNaeFxbvB6kiogi4Rr6AtFI5deFpoJJ/vp13
mvqC1Pt/JVL58GJOJmX3eAwfyUyrUsOMF0Vz9nFNFGf/dq17w/XcN4LG1c36gvkL
XN2VtkeymLVtKmAi1tiTI2w3RK1VdZL1xWtdaPDUGQBVJYcZZ9eAhPMBsUto5Lcq
jffAnbNsYU8kFCvPceZjyJZSp4LnZ7Z8aIEWLlW6R1ybtJiQwvQdgBmQ90KNyqAV
RsL0m08whEsx53nyqCr08LHcuM/w+iHmDSFsVHvR6gUGGOGI7whJTtGM+I3mR7gN
ZS2Jfw5EapSScn59EVibzvLGHjz8lPUFyy0qpc/LoQeccimY1Ta/GnuUL9kIPm8/
xJb0qdiAkJwAPB80yC5CqgTL3UgYnb5VoWfl53iMtSg1602B4EJpmIjopVFoH0+r
A9KLASK1InqneG3qo2vIRKsNkPTfp0Msmyl1pDzYIKZvxVz932Ed3uUbf/bBvQCR
4p1mbNbvIGUyn2BEkicgtep88Jqr+CfYxq80zJXNVciYBRBmIRf/NCLBw40A6pJ
uBjJzX3/gQL2LIstD7Z0nUAVoaucojiSqwYuiLE+Dw5GoZc146TCBHK6ua8FQ==
=Pk+m
-----END PGP MESSAGE-----
nmeghanathan@natarajan-VirtualBox:~$
```

Figure 18: Encrypted version of the secret message text file in ASCII format

```

nmeghanathan@natarajan-VirtualBox:~$ ls -l
total 16
-rw-r--r-- 1 nmeghanathan nmeghanathan 179 2013-02-11 05:28 examples.desktop
-rw-r--r-- 1 root root 2288 2013-02-11 07:08 mnatraj-pk
-rw-rw-r-- 1 nmeghanathan nmeghanathan 66 2013-02-11 07:13 secret-message.txt
-rw-rw-r-- 1 nmeghanathan nmeghanathan 1006 2013-02-11 07:14 secret-message.txt.asc
nmeghanathan@natarajan-VirtualBox:~$

```

Figure 19: Size of the plaintext and encrypted version of the secret message files

Question 5: As you can notice from Figure 19 above, the size of the encrypted secret message ASCII file is much larger than its plaintext version. Why is it so? Explain.

- Use the command: “sudo cp secret-message.txt.asc /home/natarajan_jsu” to transfer the encrypted message file to the default folder of the natarajan_jsu account (see Figure 20). Use the ls -l command at the receiver (natarajan_jsu) account and check the existence of the encrypted secret message file.

```

nmeghanathan@natarajan-VirtualBox:~$ sudo cp secret-message.txt.asc /home/natarajan_jsu
nmeghanathan@natarajan-VirtualBox:~$

```

Figure 20: Copying the encrypted version of the secret message text file to the receiver’s account

```

natarajan_jsu@natarajan-VirtualBox:~$ ls -l
total 12
-rw-r--r-- 1 natarajan_jsu natarajan_jsu 179 2013-02-11 05:29 examples.desktop
-rw-rw-r-- 1 natarajan_jsu natarajan_jsu 2288 2013-02-11 07:08 mnatraj-pk
-rw-r--r-- 1 root root 1006 2013-02-11 07:20 secret-message.txt.asc
natarajan_jsu@natarajan-VirtualBox:~$

```

Figure 21: Verification of the presence of the encrypted file at the default folder of the receiver’s account

8. To Decrypt the Message:

- Run the command: “gpg --output extracted-secret-msg.txt --armor -d secret-message.txt.asc” to decrypt the contents of the secret-message.txt.asc encrypted file to the named output file extracted-secret-msg.txt. It could be any named file, but preferably with a .txt extension (as we know it is indeed a text file!!) so that we can easily open the file and see its contents, as also shown next by running the cat command (see Figure 22).

```

natarajan_jsu@natarajan-VirtualBox:~$ gpg --output extracted-secret-msg.txt --armor -d secret-message.txt.asc
You need a passphrase to unlock the secret key for
user: "Natarajan M <mnatraj77@hotmail.com>"
2048-bit ELG-E key, ID 56A90FA2, created 2013-02-11 (main key ID 3AA67C15)

gpg: gpg-agent is not available in this session
gpg: encrypted with 2048-bit ELG-E key, ID 56A90FA2, created 2013-02-11
"Natarajan M <mnatraj77@hotmail.com>"
natarajan_jsu@natarajan-VirtualBox:~$

```

Figure 22: Decrypting the encrypted file at the receiver and viewing the contents of the extracted text file

Question 6: Repeat the whole of stage 3 by using RSA and RSA (option 1) for the algorithms to generate the keys for one of the two user accounts (say, in my case nmeghanathan) and retain the DSA and Elgamal (option 2) based keys that you created for the other account (in my case natarajan_jsu). Use the same e-mail addresses as you used before for the two user accounts. Capture all the screenshots of the encryption/decryption process and number them (Figures 23 through 36) as well as label them appropriately. Are you able to complete all the steps and generate all the screenshots? If you are not able to do, explain the reason? If you are able to do, explain how using two different options for the keys for the two user accounts still works? Also, were you able to associate more than one public key with an e-mail address and execute all the steps without any problem? How?