Laboratory Assignment 2: Gnu Privacy Guard (GPG)

1 Purpose

In this assignment you will learn how to setup and use gpg to sign, encrypt and decrypt files. You will also use gpg to sign and encrypt emails.

2 Reporting

To pass this assignment you have to send an encrypted and signed email to your instructor. The message should contain answers to the questions that you find in Section 5.6 (1–2 pages). In the end of this assignment, in Section 5.6.4, there are also some hints for the reporting.

3 Preparations at home

Read the chapters listed below, then read the rest of this assignment. Note that you will use GnuPG (GPG) and not PGP. The terminology for PGP is slightly different than the one for GPG in the documents below. The answers to the questions in Section 5.6 should be prepared at home before attending the lab session. Write the answers in a text editor (do not forget to save regularly) and paste the answers into the mail when you are ready to email the report.

3.1 Reading

- Section on "Public-Key Encryption Structure" within Section 2.3 in course book.
- Section 2.4 on "Digital Signatures and Key Management" in course book.
- An Introduction to Cryptography: Chapter 1
  (`http://www.pgpi.org/doc/guide/7.0/en/intro/`)
- The Gnu Privacy Handbook: Chapters 1 and 3
  (`http://www.gnupg.org/documentation/`)

4 Introduction

In this assignment, you will go through the following exercises:

1. Create your new keys with GPG
2. Encrypt and sign files
3. Configure a mail user agent (either SeaMonkey or Thunderbird) to use GPG
4. Exchange and sign keys with other student groups
5. Send signed and encrypted emails
6. Add the instructor's public key to your key ring
7. Send the final report in an email to your instructors

GPG is run by typing gpg at the prompt. Please refer to the man page for further details (`man gpg` at the prompt).

5 Exercises

The following exercises are to be done in the laboratory. Remember to save your results for the report.

5.1 Create new keys with GPG

Begin by **generating** a new keypair and add it to your key ring in gpg.

1. Choose the default key type (DSA and ElGamal).
2. Choose at least a 2048-bit key for ElGamal.
3. Set the expiry date to three months (so that the keypair will expire slightly after this course ends).
4. When GPG asks for your "Real Name", enter your name and laboration group name (First_name Last_name (Group XXX)).
5. Enter your Chalmers email account (email@student.chalmers.se).

   Note: If you don't have used IMAP towards Chalmers before, you will need to contact the IT support to retrieve the password needed to configure your IMAP client. As an option, you may also use another e-mail address of yours for this assignment, e.g., your gmail account. In that case, use this e-mail address as your email account in this step!
6. GPG then asks for a comment to include with the key. You may enter that the key is going to be used in this course.
7. To protect your key, it is very important that you enter a good passphrase. If you need help, you can look at the page `http://www.diceware.com`.

When GPG has finished creating your keys, it will print out a summary. Just above the summary you can read the following: "key marked as ultimately trusted". Make sure you understand what this implies.

You now need to generate a revocation certificate. Normally GPG prints the output to the screen. In this case you would like to save the output to a file, so please add the appropriate command line options. You may select any reason for the revocation. As mentioned in the GPG manual, the revocation certificate needs to be kept in a safe place. Many people print it and place it into a safety vault. For this course, it is enough if you make sure that the revocation certificate file only can be read by you (`man chmod`).

Finally, you are going to edit the key and add the name (user IDs) and email of the other member of your group. Make sure that the user ID containing your lab group name is marked as the **primary** key when you have finished.

Hint: Selected keys or UIDs are indicated by an asterix. You toggle the selection of UIDs with the command `uid`.

When you're done, you should have similar output to the one shown in Figure 1. You may need to quit and restart GPG to ensure all fields are updated.

Note: If your output does not look like the one in Figure 1, you need to ask the instructors for help. Note the two user IDs and the placement of the period: "(1).".

```
Command> list
pub  1024D/2D5362CE  created: 2007-08-29  expires: 2007-11-27  usage: SC
                     trust: ultimate      validity: ultimate
sub  2048g/4A6F78DB  created: 2007-08-29  expires: 2007-11-27  usage: E
[ultimate] (1).  Instructor Fred (Instructor in Computer Security) csec140 <email1@...>
[ultimate] (2)  Instructor Amy (Instructor in Computer Security) <email2@student...>

Command>
```

Figure 1: Example of multiple UIDs in gpg

5.2 Encrypt and sign files

Download a copy of the document "Why do you need PGP?" written by Phil Zimmermann from `http://www.pgpi.org/doc/whypgp/en/`. Use "`Save as`" to save it as text with the name `whypgp.txt`.

1. Encrypt and sign this file. Remember that you also need to specify the recipient. In this case, it is your own ID as you want to be able to decrypt the file later.

2. You now have a file called `whypgp.txt.gpg` in your directory. Look at this file with the command `less whypgp.txt.gpg`

3. Run gpg the same way again, but this time also include the option `armor`.

4. Look at the new file created by gpg called `whypgp.txt.asc`. How is it different from before? Why do you think the `armor` option is useful?

5. Decrypt the files again. If you want gpg to save the output to a file, you need to include the command line option `--output whypgp.decrypt`.

6. Compare the decrypted file with the original (for example with `diff whypgp.txt whypgp.decrypt`). Are the two versions identical?

5.3 Configure your mail user agent to work with GPG

5.3.1 Thunderbird

Start the Thunderbird mail client by typing `thunderbird &` at the command prompt, and **not** through the menu system. You now have to configure your mail client. Because Chalmers now use Outlook live as mailserver for students, we cannot give you complete information on how to configure the mail client. You will have to find out some information yourself by logging on to your web-mail through the Student Portal. The information you need to find out is name, portnumber and encryption method for the IMAP server that you should use for receiving mail. You also need to find the same information about the SMTP server that you should use for sending mail.

Note: If you are using your own e-mail account, you need to find the IMAP address and port, SMTP address and port, plus the encryption used for each of these protocols. This information is needed for configuring Thunderbird.

When you have found the needed information you can follow the instructions at `http://www.cse.chalmers.se/edu/course/EDA263/Mail/mail.html`. This instruction is actually written for Seamonkey mail but it should be useful also for Thunderbird as there are only minor differences in the configuration.

5.4 Exchange and sign keys with other students

Before you can encrypt an email for your friends, you need to acquire their public key (remember that you use their public key to encrypt while they use their private key to decrypt). As you know from the required reading for this assignment, gpg uses something called the web of trust where you verify and sign other people's keys.

1. Export your public key (preferably in text format) to a file (with the name `myPublic.gpg` or `myPublic.gpg.asc` depending on the format used) using gpg.

2. Send the public key to some of your friends in the laboratory.

3. When they send you their key, you first import it into gpg (adding it to your key ring).

4. To ensure that the key you received actually belongs to your friends (remember that there might be man-in-the-middle attacks with emails), you should check its fingerprint. List the fingerprint of their key, and ask them to verbally read it off to you.

5. If the fingerprints are identical, you sign their key.

6. Finally, you export their key into a file and email it back to the owners so that they can import it back into their key ring (thus updating the original key with your signature that vouches for the key validity).

You should also add the *owner trust* (as opposed to the key validity concept) to the key. Edit the key you just added to your key ring and add appropriate *owner trust* with the command `trust`.

To pass this lab, you need to have **at least two groups** sign your key!

Note: Many students tend to fail to have at least two keys from other groups in their keyring. Verify that you have a similar output to Figure 2 with the proper command.

```
uid  Dharma <dharma@ce.chalmers.se>
sig!3        56520905 2006-08-29  [self-signature]
sig!         471C384B 2006-08-29  Alice <alice@ce.chalmers.se>
sig!         7126E436 2006-08-29  Blake <blake@ce.chalmers.se>
sig!         82EC4BE1 2006-08-29  Chloe <chloe@ce.chalmers.se>
```

Figure 2: Example of a key signed by three users

5.5 Send encrypted and signed email

Now try sending emails to your friends in the laboratory. First, (only) sign the message and let your friends verify your signature. Next, encrypt and sign the message. Let your friends verify and decrypt the email.

5.6 Send message to your instructors (Reporting)

In the final part of this lab, you should send an encrypted and signed email to your instructors.

5.6.1 Fetch the instructors' public key

To be able to encrypt your email to your instructors, you need their public key.

1. Check the homepage for this assignment in PingPong for instructions about the public key.

2. Fetch the public key from the key server and add it to your key ring. You may either download the key directly with gpg, or use the web interface to save the public key to a file that you then import.

3. Verify that the signature of the key you just imported corresponds to the fingerprint we have listed on the course homepage.

5.6.2 Verifying your key

Make sure that your key has at least two user IDs, and has been signed by two other lab groups.

5.6.3 Sending the report

Write an *encrypted and signed email* to your instructors. The e-mail address is given at the assignment homepage in PingPong. Make sure that the subject of the email is correct (*Lab 2, group XXX*). Please avoid using any attachments, and include all the information directly into the email. The message (1–2 pages) must contain answers to the following questions (**do not forget your key in Q9**):

1. What are your names, lab-group, and civic registration numbers?
2. What is the fingerprint of your key?
3. What mail client did you use for the lab?
4. Will you use gpg in the future? Please specify your reasons.
5. What is the purpose of signing in terms of the "CIA" (Confidentiality, Integrity, Availability)? Motivate your answer. Explain how signing works in PGP? Why is a hash function used?
6. What is the purpose of encryption in terms of the "CIA"? Motivate your answer. Explain how encryption and decryption works in PGP? Make sure that your answer includes how PGP combines the best features of both conventional and public key cryptography.
7. In the lab you validated the keys by asking the owners verbally for the fingerprint. Explain how the trust system (*web of trust*) in GnuPG works and how you validate a key belonging to a person you do not know personally. Your answer must distinguish between *(owner) trust* and *(key) validity* (sometimes called validity trust). You should also include the conditions for when you start to consider keys from foreigners valid.
8. Consider the example shown in the Figure 3 below. Assume you have (at least) five keys on your key ring. You have personally signed the keys of Alice, Blake and Chloe. They, in turn, have signed the key of Dharma. Finally, Dharma has signed the key of Elena. From the figure, you can see the current *owner trust* as well as the *key validity* (calculated trust value). To be succinct, we have marked these entries as -/-, where the left entry stands for owner trust and the right entry key validity.

   For example, consider Alice. As you have personally signed her key, it is **f**ully trusted (i.e. you know that the person Alice owns this key). Furthermore, you also know that Alice is quite messy and has a history of signing other people's keys without the proper validation. For that reason, you have assigned the *owner trust* of Alice to **m**arginal. The same goes for Blake. However, Chloe is a new acquaintance and you do not know whether she is careful when validating keys.

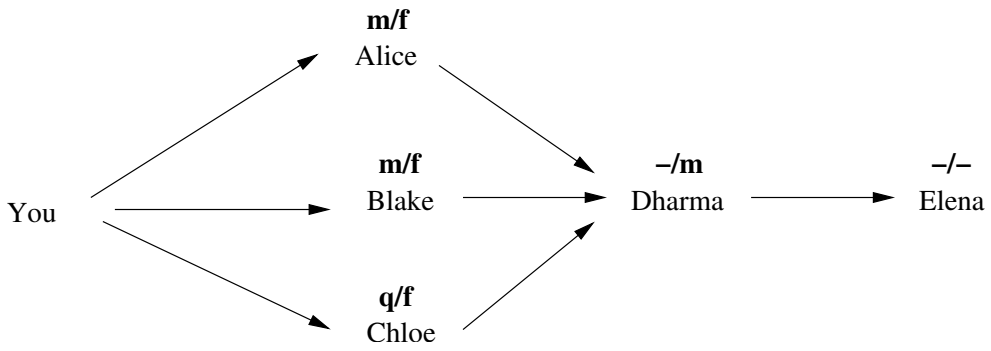

Figure 3: Example of a web of trust. '$A \longrightarrow B$' indicates that A has signed B's key.

Hence, you put "Don't know" (q) for owner trust of Chloe. Depending on your version of GPG, this may be listed as *trust: undefined, validity: full*. You have not entered any owner trust for either Dharma or Elena, hence the – in the figure (may be marked as *unknown* in gpg).

(a) You know Dharma, and you now enter that you trust her fully. If you list the key for Dharma, it would now show: f/m (*trust: full, validity: marginal*). What would be listed for Elena? Answer using the form: Elena trust: x, validity: x

(b) Time passes, and you realize that Chloe is actually somewhat trustworthy. For that reason, you change the owner trust of Chloe to marginal (Chloe: m/f). What would now be listed for Elena? We assume the changes done in (a) are still in effect.

Hint 1: If you are uncertain of the answer, you can download the file from the Documents directory in PingPong (lab2.keys). Import all the keys in this file, sign Alice, Blake, Chloe and set the proper owner trust levels. Then you can just print the values for Elena.

Hint 2: In the offprint-collection for the course, there are two papers describing cryptography: *An introduction to cryptography* and *The GNU Privacy Handbook*. The two papers give different information regarding the number of marginally trusted keys that need to sign a key to make it valid. For Q8, lab2 you should use the information provided in *The GNU privacy handbook*.

9. What is your public key (in `ASCII`-format)? Use the menu option *Attach My Public Key* in Enigmail/OpenPGP to include your key with the email.

   You can also manually insert the key into the email. First, you need to export a fresh copy of the key (with the armor option) to a file. Second, include the contents of this file into the email. Make sure that there is a blank line between your answers above and the start of the preamble of the key block:
   `-----BEGIN PGP PUBLIC KEY BLOCK-----`

5.6.4 Some additional hints for the Lab 2 report

Remember that your answer MUST be in your own words. You cannot copy text from public sources, but you need to write the report in your own words and cite the public sources you use to collect information.

Q5: Note that this question has 3(!) parts:
(A) CIA: You should give your answer + a motivation.
(B) How does signing work in PGP: Give a step-by-step list (1) ... (2) ... (3)... Also explain how the signature is checked by the recipient.
(C) Why is a hash used: explain why a hash is used. For (C): Note that the hash is not used to ensure that the data has not been tampered with. The hash algorithms are public, so anyone can create a hash of a text. From your answer in (B), you should have one separate step for the hash (and this is different from the step describing the method that ensures that nobody can tamper with the message content). Thus, for (C) you need to specify WHY hashing is one step of your answer in (B).

Q6: Note that this quest has 3(!) parts:
(A) CIA: You should give your answer + a motivation.
(B) Explain the steps of encryption / decryption: Give a step-by-step list, (1) ... (2) ... (3) ... Ensure that you also explain the decryption process and what key is used for each step.
(C) From your answer in (B), you should realize HOW PGP combines the best features of both conventional and public key cryptos. List the reason the conventional crypto is used, and then list the reason the public key crypto is used.