Laboratory Assignment 1:
Gnu Privacy Guard (GPG)

For all the assignments in this course, you are expected to work home and only book a lab slot if you are stuck and require a TA's assistance or you are required to demonstrate your solution.

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 5.5. You also need to answer the quizz in Canvas as stated in Section 5.6.

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 5.5 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: Chapters 4 and 5 in the offprint.
- The GNU Privacy Handbook: Chapters 1 and 3
(`https://www.gnupg.org/gph/en/manual.pdf`)

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 (Thunderbird[1]) 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

For the following exercises please remember to save your results for the report.

5.1 Create new keys with GPG

Make sure GPG exists in your system, for Windows you may need `https://www.gpg4win.org/`. Begin by **generating** a new keypair (**gpg --full-generate-key**) and add it to your key ring in gpg.

1. Choose the key type (RSA and RSA)[2].
2. Choose at least a 4096-bit key[3].
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).
5. Enter your student email account (cid@student.chalmers.se or gus•••••@student.gu.se).
6. GPG then asks for a comment to include with the key, keep it short like your group number.

[1] You can choose a different one if you so want, but we won't be able to provide you propper support then.

[2] You could also use an ECC/ECC key set if you so wanted. These are less widely supported and, therefore, require the **--expert** option being used too

[3] If you chose ECC keys choose a curve with at least 256-bits security, we recommend Curve 25519 or any of the NIST curves: P-256, P-384 or P-521)

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 your UID 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.

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

```
Command> list
pub  1024D/2D5362CE  created: 2022-01-29  expires: 2022-11-27  usage: SC
                     trust: ultimate      validity: ultimate
sub  2048g/4A6F78DB  created: 2022-01-29  expires: 2022-11-27  usage: E
[ultimate] (1).  Student Socrates (Group X) <CID@student.chalmers.se>
[ultimate] (2)  Student Plato (Group X) <CID@student.gu.se>

Command>
```

Figure 1: Example of multiple UIDs in gpg

5.2 Encrypt and sign files

Download a copy of the document "Why I wrote PGP" written by Phil Zimmermann from `https://www.philzimmermann.com/EN/essays/WhyIWrotePGP.html`. 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 Exchange and sign keys with other students

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

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 classmates 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 classmates (remember that there might be man-in-the-middle attacks with emails), you should check its fingerprint. List the fingerprint of their key, either take a picture and verify that it matches yourself or ask them to verbally read it off to you so you can verify it.
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).

```
uid  Dharma <dharma@student.chalmers.se>
sig!3        56520905 2022-01-29  [self-signature]
sig!         471C384B 2022-01-29  Alice <alice@student.chalmers.se>
sig!         7126E436 2022-01-29  Blake <blake@student.chalmers.se>
sig!         82EC4BE1 2022-01-29  Chloe <chloe@student.gu.se>
```

Figure 2: Example of a key signed by three users

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!

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.

5.4 Sending Encrypted Emails

For this task you need to use an e-mail client agent like Thunderbird[4]. In Appendix A we provide a generic guide to set up Thunderbird. Please use these instructions:

1. Configure IMAP and SMTP for your email account. Chalmers IT Support provides a guide at `https://chalmers.topdesk.net/tas/public/ssp/content/detail/knowledgeitem?unid=2e6baf3e279f4e438f4b62416501dcbb`. NOTE for Chalmers students: Your email address is cid@student.chalmers.se, your username is cid@chalmers.se. For IMAP use port 993 and SSL/TLS. For SMTP use port 587 and STARTTLS (see Figure 3).

2. Import the public keys from the groups that sign your key inside Thunderbird.

3. Import your secrete-keys inside Thunderbird.

4. Make sure your import the correct keys, if your key is signed by others Thunderbird will try to verify, if it fails will not mark your key as personal!

5. Send an encrypted e-mail to one student that verified your key.

5.5 Send message to your instructors

In the final part of this lab, you should send an **encrypted and signed** email to your instructors. In the e-mail you should only provide answers to the following questions:

1. What are your names, lab-group?

2. What is the fingerprint of your key?

3. Please attach to this e-mail your public key (with the other group signatures) so we can verify it.

Correction of these submissions is mostly automated so please avoid adding unnecessary information to reduce the risk of errors.

[4] You can choose a different one if you so want, but we won't be able to provide you propper support then.

5.5.1 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 Canvas 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.

5.5.2 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 Answer the quiz

Questions related to the lab are included in the Home Assignments as a separate section. These Home Assignments, like all others, must be completed **individually**.

A Appendix: Configure Thunderbird

A.1 Configure your mail user agent to work with GPG

A.1.1 Install the necessary software

You may install and configure your favorite email manages, but we provide information for Thunderbird. You need to download the latest version of Thunderbird `https://www.thunderbird.net/en-US/`. The software supports Mac OS, Windows, and Linux (Ubuntu may have it by default).

Make sure you have latest version of Thunderbird.

A.1.2 Thunderbird

Start the Thunderbird mail client (e.g., by writing Thunderbird on the Windows search bar using the Windows menu).

A.1.3 IMAP and SMTP access

If you have already configured access to your e-mail account through IMAP and SMTP you can skip to the next subsection..

If this is the first time you are using Thunderbird you will be greeted by the wizard for adding an e-mail address. Chalmers IT support provides a guide for this at `https://chalmers.topdesk.net/tas/public/ssp/content/detail/knowledgeitem?unid=2e6baf3e279f4e438f4b62416501dcbb`. One way is to cofigure your settings manually (see Figure 3). **Your email address and your account name is not the same (!)**. You need to login with the follow settings, email: **cid@student.chalmers.se**, username: **cid@chalmers.se**. For IMAP use port 993 and SSL/TLS. For SMTP use port 587 and STARTTLS (see Figure 3).

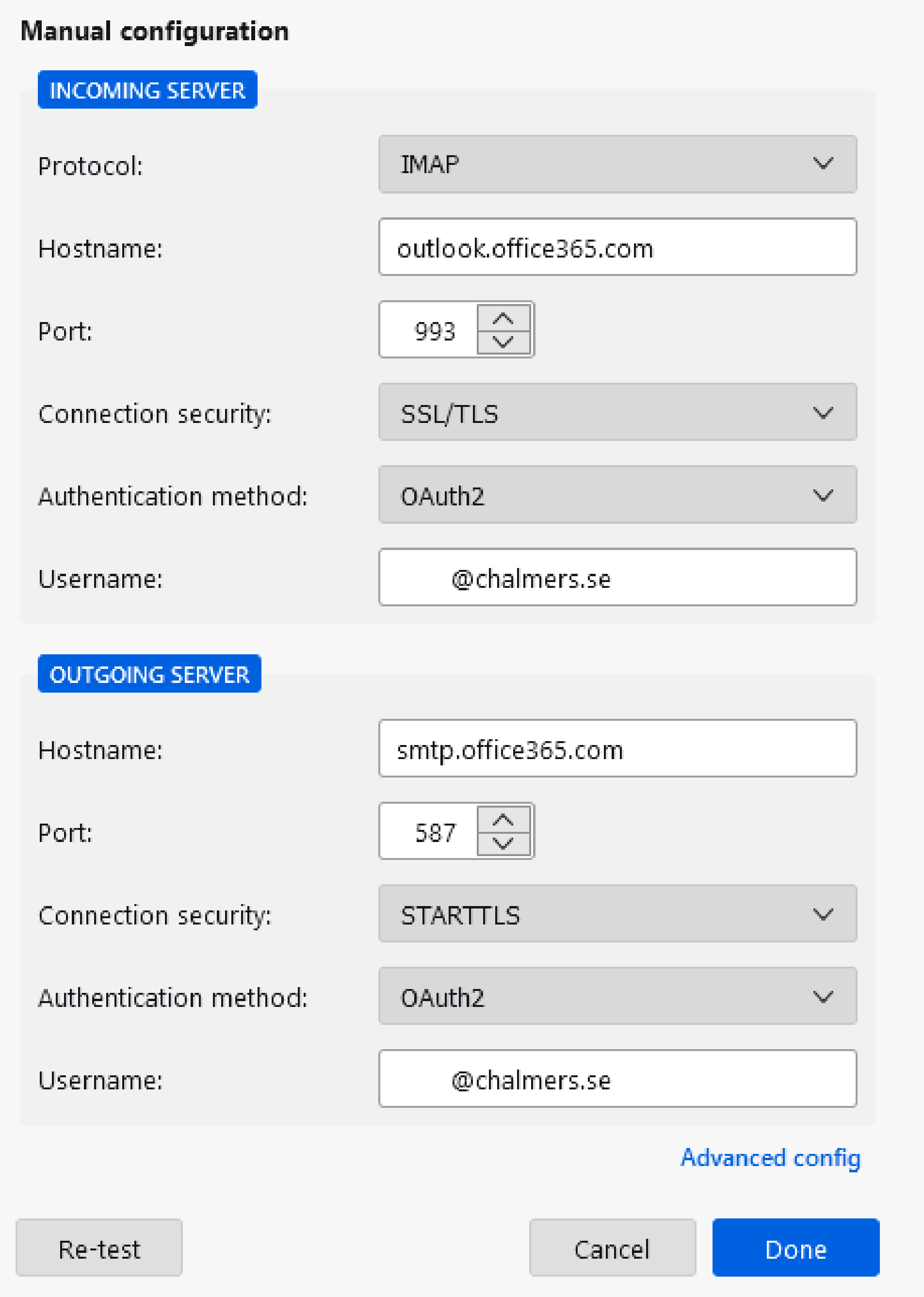
Before continue, validate your settings works! (e.g., send a test email and reply with another email account).

A.2 Import GPG keys

For this step, you need the two public keys of the other groups member that signed your key. Also, you need to export your private keys from GPG: **gpg --export-secret-keys --armor >my-secret-keys.asc** . Please navigate to your settings and end-to-end encryption as shown in Figure 4. Using Add key... start by importing first the public keys of other group members. Next, import your private key and make sure you mark as personal.

A.3 Sending Encrypted emails

After you have successfully imported your pair-keys you can send an encrypted email by clicking on "Write" and in the tab "Security" (See Figure 5).



Manual configuration

INCOMING SERVER

Protocol: IMAP

Hostname: outlook.office365.com

Port: 993

Connection security: SSL/TLS

Authentication method: OAuth2

Username: @chalmers.se

OUTGOING SERVER

Hostname: smtp.office365.com

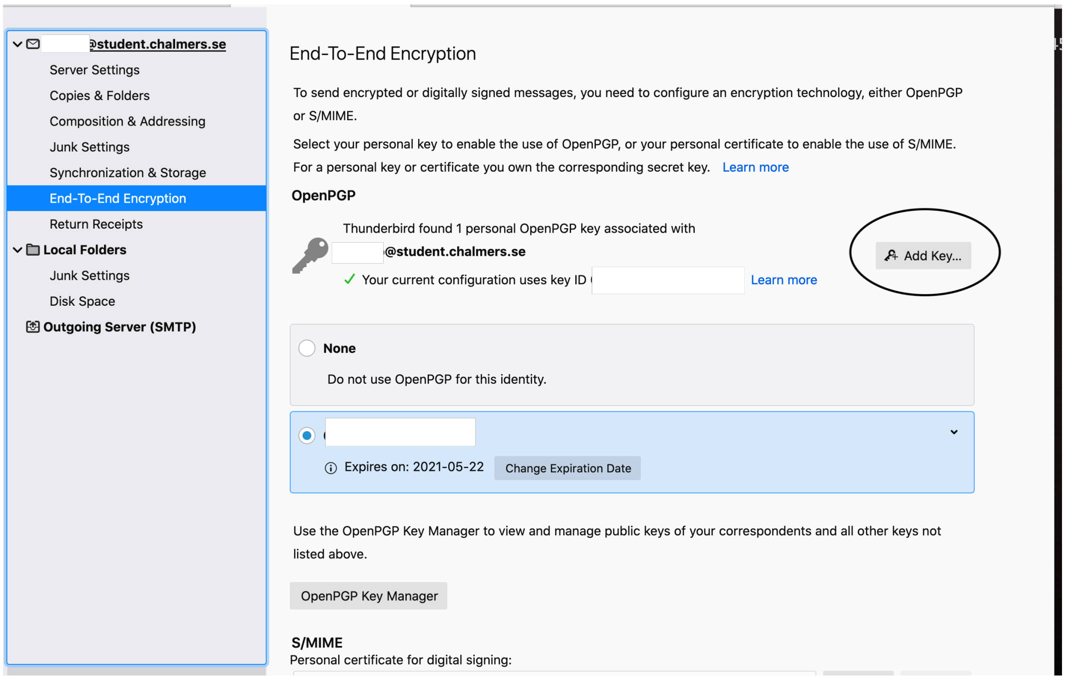
Port: 587

Connection security: STARTTLS

Authentication method: OAuth2

Username: @chalmers.se

Figure 3: Thunderbird settings for your student account.



End-To-End Encryption

To send encrypted or digitally signed messages, you need to configure an encryption technology, either OpenPGP or S/MIME.

Select your personal key to enable the use of OpenPGP, or your personal certificate to enable the use of S/MIME. For a personal key or certificate you own the corresponding secret key. Learn more

OpenPGP

Thunderbird found 1 personal OpenPGP key associated with @student.chalmers.se

✓ Your current configuration uses key ID Learn more

○ None

Do not use OpenPGP for this identity.

● Expires on: 2021-05-22

Use the OpenPGP Key Manager to view and manage public keys of your correspondents and all other keys not listed above.

S/MIME

Personal certificate for digital signing:

Figure 4: Add GPG keys inside Thunderbird.

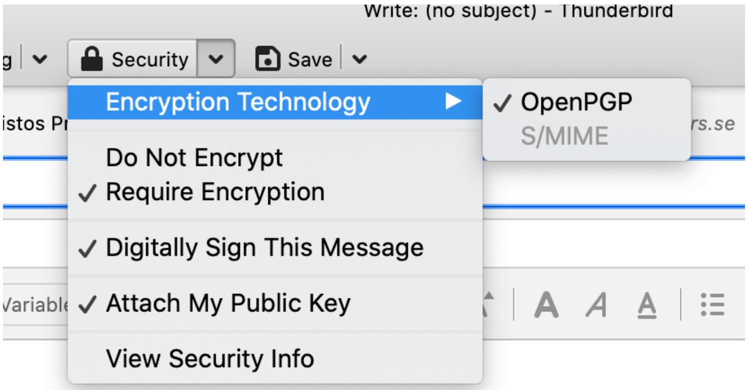
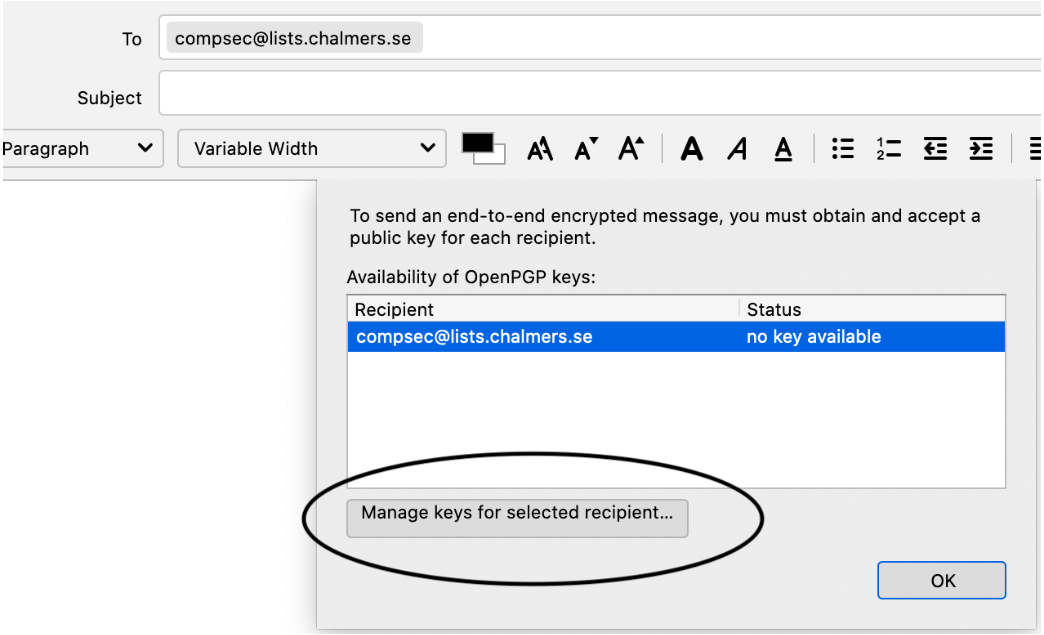


Figure 5: Send encrypted emails.



To compsec@lists.chalmers.se

Subject

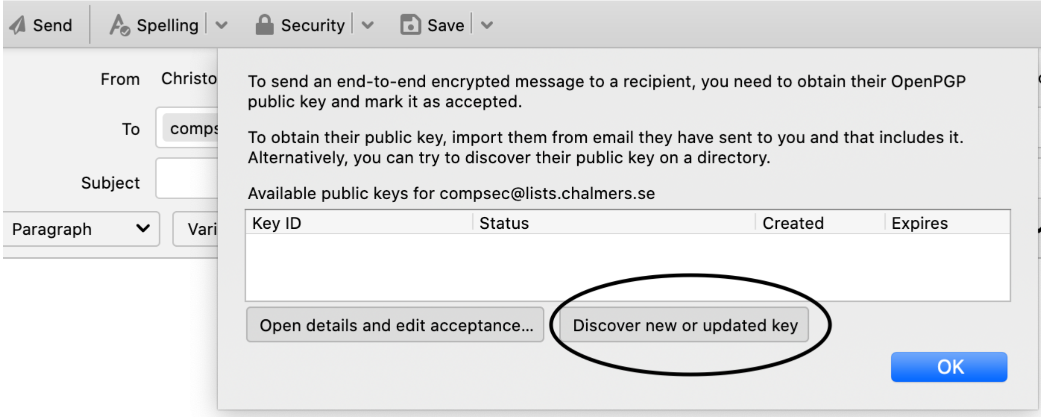
Paragraph Variable Width

To send an end-to-end encrypted message, you must obtain and accept a public key for each recipient.

Availability of OpenPGP keys:

Recipient	Status
compsec@lists.chalmers.se	no key available

Figure 6: Import Public-Key of the Course Step-1



To send an end-to-end encrypted message to a recipient, you need to obtain their OpenPGP public key and mark it as accepted.

To obtain their public key, import them from email they have sent to you and that includes it. Alternatively, you can try to discover their public key on a directory.

Available public keys for compsec@lists.chalmers.se

Key ID	Status	Created	Expires

Open details and edit acceptance...
Discover new or updated key
OK

Figure 7: Import Public-Key of the Course Step-2