# NLUUG Keysigning Party

Autumn conference 2022-11-29 Utrecht

## Introduction

Over the years NLUUG has hosted various PGP/GPG keysigning parties, and the program committee of the autumn conference thought it would be a good idea to reinstate this tradition once more.

## What is a keysigning party?

A keysigning party is a get-together with PGP/GPG users for the purpose of meeting other PGP/GPG users and signing each other's keys. This helps to extend the "web of trust" to a great degree. Also, it sometimes serves as a forum to discuss strong cryptography and related issues.

## What must I do to join this keysigning party?

If you already have a key, then the details on what to fill in are listed below. If you do not yet have a key, please see the How do I prepare section below.

All you need to send to the enlist email address is your fingerprint. This can be found in the output of the command `gpg --fingerprint` *`youremail@domain.com`* and looks like `AB89 08EF DCEE C230 C90E  C746 4E98 5C76 7F64 23B4`.

You also **must** ensure that your keys are available on the `hkps://keys.openpgp.org` keyserver. You can do this with `gpg --keyserver hkps://keys.openpgp.org --send-key` *`keyid`* where *`keyid`* is your key ID. The (long) key ID consists of the lowest 64 bits of your key fingerprint. Sometimes the key ID gets prefixed by `0x` since it is a hexadecimal value.

# What do I need for this party?

1. Physical attendance (*id est* you must be physically present at the party).
2. Positive picture ID:
    - two piece of ID are recommended;
    - at least one should be government issued.
3. Your key ID, (hexadecimal) fingerprint, key type, and key size from your key.
4. A pen/pencil or whatever you would like to write with.

# How do I prepare?

## Create a keypair

If you do not already have keys, follow the instructions on the `gpg` manual page, to create a keypair.

## Register your fingerprint

Please send the fingerprint to the following enlist email address to register: `keysigning@nluug.nl`. If possible sign your email using the key you are registering.

## Print a copy of your fingerprint and bring it to the conference!

Run `gpg -K --fingerprint youremail@domain.com` and print the results. To save trees you can ofcourse also store it in digital form.

# What happens at the party?

Note that all of this will be explained at the party. But you may choose to familiarise yourself with the basic idea.

First, each person will get a piece of paper with the fingerprint of every key that was sent to the above mentioned email address, and some checkboxes next to each one.

Then, each person will read off their fingerprint **from their own personal copy** of their fingerprint that they **brought with them from their private key**. As they do this, each person will verify that the fingerprint on the list they received is in fact valid.

We then get in a (big) "conga-line." This involves splitting into two equal lines, and having these lines face each other. You then verify the identity of the person in front of you. This should include seeing **official identification**. How much verification you need to **state to the world you believe this person to be the name on their key** is up to you. It is common to require two forms of ID at least one of which is picture ID and one of which is government ID.

Once everyone is ready, everyone shifts down one and repeats the process. This whole thing is repeated until everyone has verified everyone. Checkboxes are provided next to each key on your list to make it easy to keep track of who you have verified.

## What happens after the party?

This is documented here for reference after the party. After the party, you return to your laptop, mainframe or workstation and sign and deliver keys. This is quite time-consuming, so set aside some time within a week of the party to do this.

NLUUG will publish a keyring with everyone's key to make things a bit easier for everyone, but since every key should be available on the `keys.openpgp.org` keyserver you can lookup and import the keys yourself as well. From here you can verify those keys, sign them, and verify email addresses.

### Import the keyring

The keyring location will be specified here, after the party. Download it to a file. You can either import it into your keyring, or use it with `--keyring` *`file`*.

### Verify the key

For each key, verify the key you have is the same as the key on the paper you've verified: `gpg --fingerprint` *`keyID`*. Please verify the **complete fingerprint** returned.

### Sign the key

You can sign each of the keys you have verified manually using the following command: `gpg --ask-cert-level --sign-key` *`keyID`*.

You will be asked what level you want to sign it at:

- **Level 3** is if you have verified their fingerprint, ID, and email carefully enough that you feel confident **stating publicly** that you **personally vouch** for the fact that the person with the name in question owns the key and the email address in question.
- **Level 2** is "casual checking" - perhaps they only had one form of ID, or they had two IDs that you did not feel were up-to-snuff, or something else that makes you less than 100% sure.
- **Level 1** is, in many opinions, useless, as it says you have done no checking at all.
- **Level 0** is also useless, as it says you decline to answer (which would probably be better served by adding local trust to the `trustdb` instead of signing their key).

For every key you sign you will have to select a level and also type in the passphrase to your key (to unencrypt it, so it can be used to sign the key in question).

## Send the key securely to the owner

This describes an easy (and reasonably secure) way to verify the email address, and return the signed key to its owner.

For each UID (name and email address) on a key for which you have verified the fingerprint you sign the key. Many times there will only be one name/email UID assigned. Once you have signed all verified UIDs you export the signed key to a file.

You then **encrypt-email** this to the email address you have verified. This ensures that only the owner of the private key, most probably the owner of that email address, can decrypt that email. This method is sufficient for the vast majority of even paranoid PGP users.

## PIUS: The PGP Individual UID Signer

Signing keys after a PGP Keysigning party can take a lot of time. Further, it is relatively difficult to do the right way: signing each UID separately and emailing it off is not something the standard (`gpg`) tools make easy. PIUS is a tool developed to solve both of those problems and make signing keys easier and faster.

PIUS can be found on Github at https://github.com/jaymzh/pius/.