Encrypting Messages with PGP

Introduction to sending secure encrypted messages

CATALOG SHEET

Internal Document of the Revolutionary Technical Committee
Work: Encrypting Messages with PGP
Pages: 10 whole, 8 work

Internal Manual

CSWW
A-00-001
revteccom.tk/csww.html

What is PGP?

PGP stands for Pretty Good Privacy, it is the name of a program originally created in 1991, but nowadays, the acronym is used to refer to any software that conforms to the OpenPGP specification. PGP is a way to encrypt and authenticate messages and files, mostly for transferring data over a network where you know someone will have access to the contents of a message.

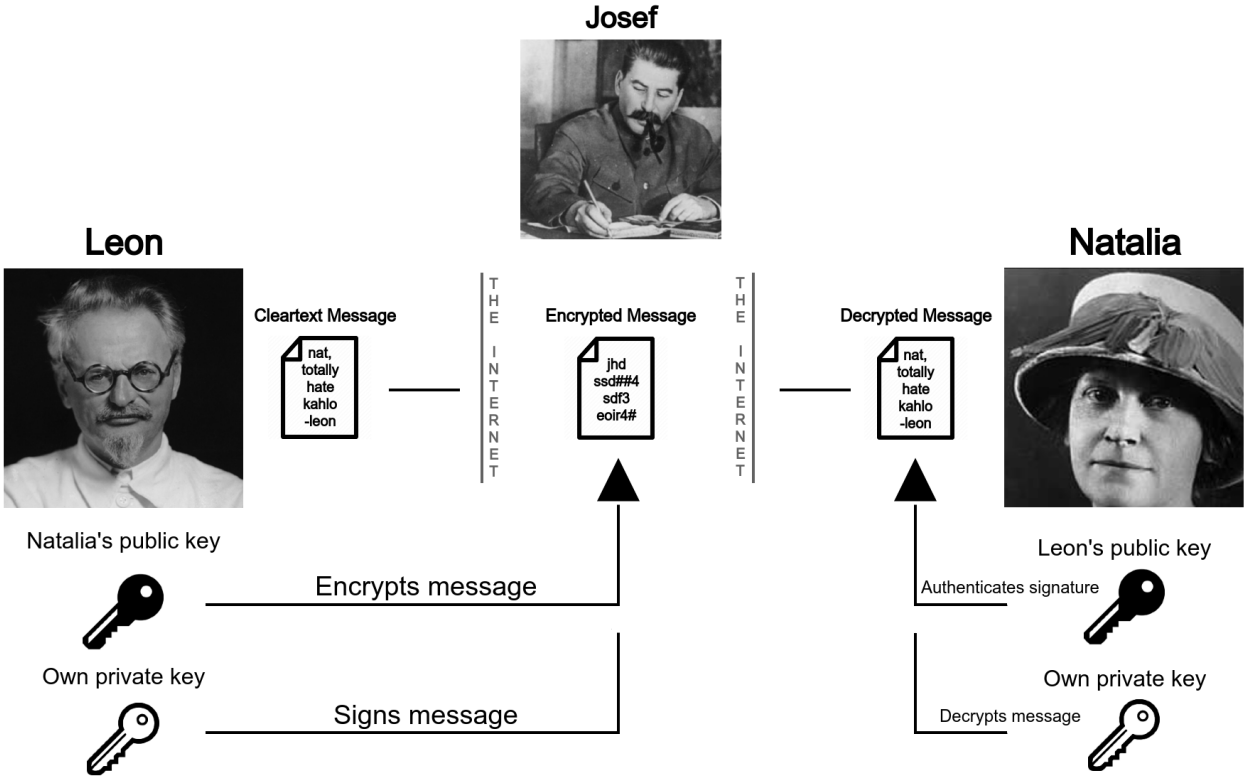
OpenPGP stands the closest to a military-grade encryption tool, and its algorithms are currently unbreakable for standard supercomputers. Even a full-fledged quantum computer (which, as far as we know, does not yet exist) may still take years to break them. If you and the person you wish to communicate follow all the steps within this manual, you can be positively assured that your communications will be secure.

So, let's exemplify the standard use for PGP communication: encrypting an e-mail. Envision two people: Leon and Natalia. Now, Leon wants to send a message full of specific compliments to Natalia, and they both want to make sure nobody can see it. In that case, of course, Leon would want to use to encrypt the message before sending it.

There are five basic steps to actually encrypt and send a message using PGP:

1. First, Natalia would have to create a public and private key, called a "key pair".
2. Natalia would share her public key with Leon, ensuring she keeps her private key safe.
3. Leon would take the public key he got from Natalia and encrypt the message he wants to send over to her.
4. Leon would cryptographically sign the message with his own private key, proving its authorship, and send over his public key to Natalia.
5. Leon would send the message over to Natalia, and she would decrypt it with her private key.

Below is a visual illustration of the dynamics of PGP:



Now, if Josef intercepts Leon's message, he will only see a garbled mess. Even if Josef, for example, sends a man with an icepick to Leon's house to extort him for his private key, he will not be able to decrypt the message. Instead, he would have to acquire the key from the person the message is directed towards, so, in this case, he would need Natalia's private key.
It's also worth noting that, while Natalia can't know a message came from Leon if he didn't sign it, Leon can also leave it unsigned to deny responsibility for the message's contents. So, if Leon sends a scandalous unsigned message to a Josef who is posing as Natalia and has her private key, he can claim the message isn't his because he has left it unsigned, and thus frustrate Josef's attempt to make up a media outrage.

How to install GnuPG and Kleopatra (GNU/Linux)

Install Kleopatra via your favorite package manager. On Ubuntu, Mint or any Debian-based distribution, it should be under the name "`kleopatra`". It's also under that name on Fedora/RHEL repositories and on the Arch repositories. For Arch distributions that do not use the standard repositories, there is a slightly outdated version on the Arch User Repository called "`kleopatra-git`".

Distribution	Installation Command
Debian	`sudo apt install kleopatra`
Fedora/RHEL	`sudo dnf install kleopatra`
Arch	`sudo pacman -S kleopatra`

It is recommended you run Kleopatra under the X Window System.

How to install GnuPG and Kleopatra (Windows)

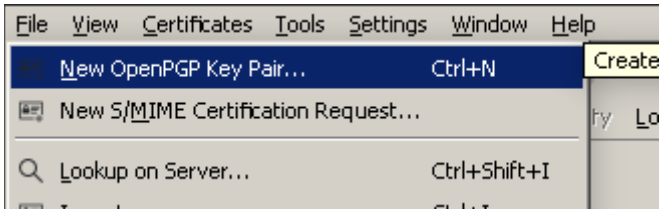
WARNING: Using Windows is not recommended at all! Your messages may already be compromised by Microsoft's telemetry measures – while PGP's one-way encryption system is fine for sending messages, **you should reasonably assume Microsoft will provide a secret backup of your private key and passphrase at the request of any government agency**, so receiving messages is not safe.

Download the official gpg4win package from https://www.gpg4win.org/. Windows' User Account Control should automatically authenticate the code signature for you. If it marks the software as coming from an unknown source, this means the binary you downloaded is unsigned – check the gpg4win news section to see if that's something you should worry about.

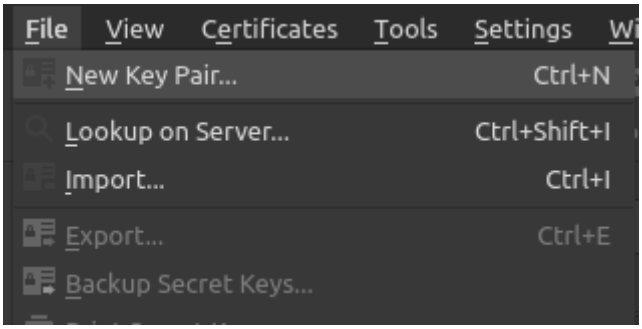
Once you have started the installation, you will be met with the standard install wizards you see in many Windows applications – there's no secret here, just click "Next" until the installation is completed.

How to create a key pair

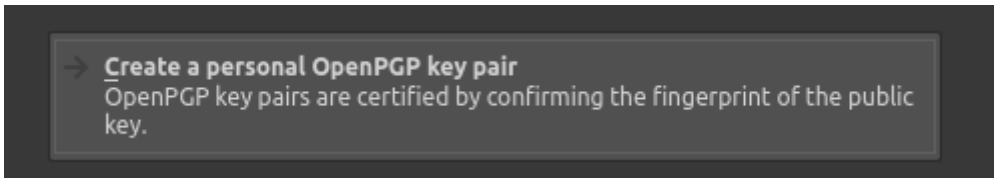
Creating your public and private key is very simple: In the Linux version, click the “File” menu, click “New Key Pair”, and select “Create a personal OpenPGP key pair”. In the Windows version, click the “File” menu, and select “New OpenPGP key pair…”.



Process on Windows – Single Step

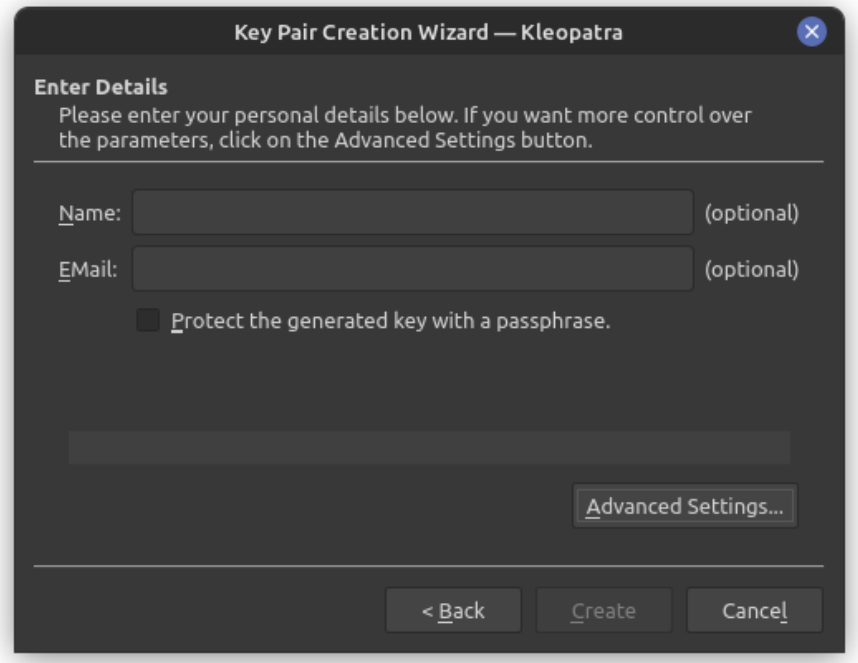


Process on Linux – Step 1



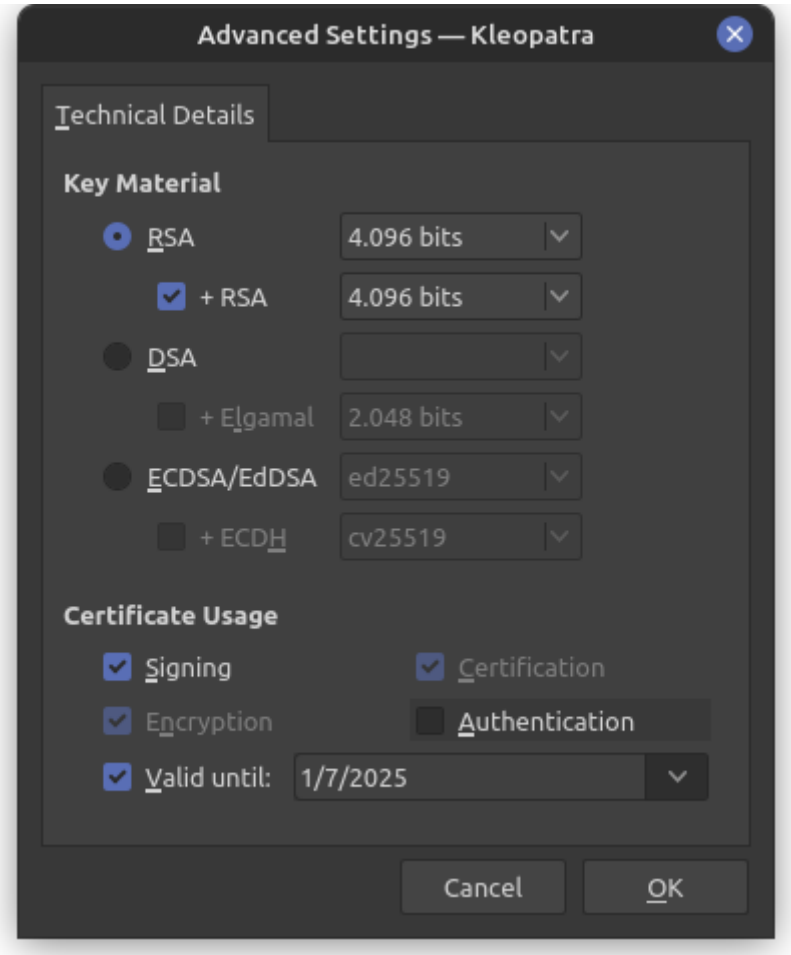
Process on Linux – Step 2

After following those steps, you will see a menu like this:



Put your desired name and e-mail address. **Remember, PGP is pseudo-anonymous, encrypted data for which you are the receiver will have your key metadata attached to it**. So, choose a pseudonym, and maybe leave the e-mail field empty, unless you benefit from such metadata being verifiable. You may also choose to protect your key with a passphrase – that's up to you, and is **highly recommended**.

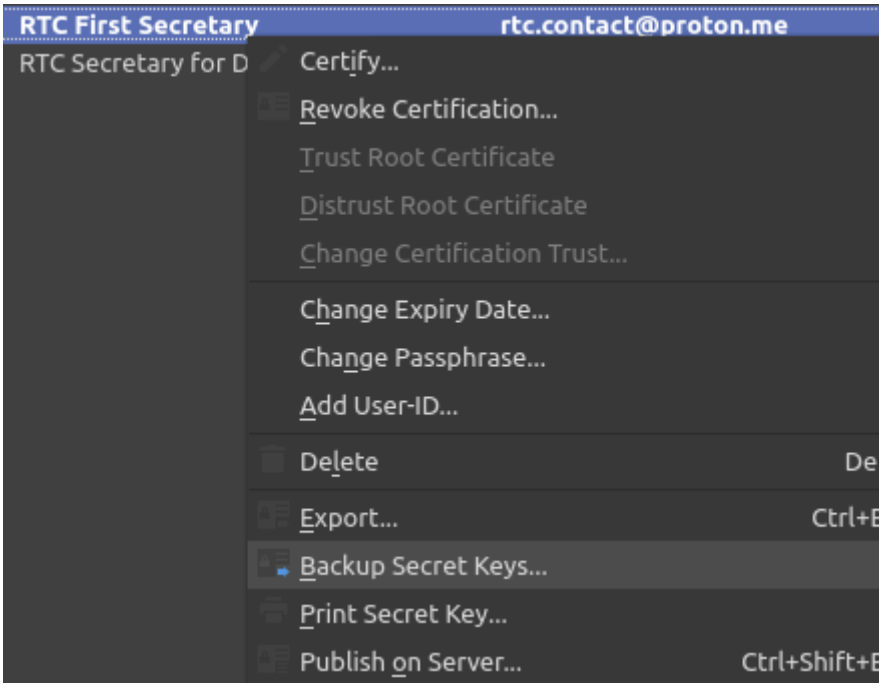
Next, click the "Advanced Settings" menu, and ensure you have selected the RSA + RSA encryption type, both at 4096 bits. You may also select an expiry date for your key, or no expiry date at all – since you won't be around forever, it's recommended your key isn't around forever, either. On average, select an expiry date 5-7 years from the current date for your personal key. Once a private key expires, it may be renewed, and new versions of the public key may be distributed. Overall, you should have something that looks like this:



Click "OK", **and before clicking "Create"**, think up a passphrase. Do **not** store the passphrase on your computer!

It is highly recommended you make a backup of your secret key on inexpensive and easily-destroyable media, such as a small flash drive or even paper. This will be presented as an option after key creation in Kleopatra on Linux. **Do not, under any circumstances, make online backups, also, you should store backups DIRECTLY into flash drives**. **You should only print your private key if you have direct access to the printer and can disable its Wi-Fi functionality**.

To export your private key, go to the “Certificates” section (the one Kleopatra is open on by default), select your key, right-click it, and select “Backup Secret Keys...”



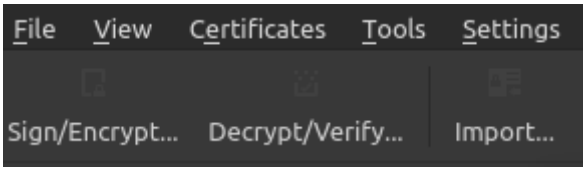
Example with option highlighted

A save menu will appear. Save the key on your preferred external media, and store it under a floor tile, at the bottom of a drawer, or wherever an ill-intentioned party may have a hard time finding it.

Importing a public key

Public key files can be downloaded in many places. Some places, called “keyservers”, are giant repositories of public keys. For the sake of following this example, you should import the key of the office of the First Secretary of the Revolutionary Technical Committee, which can be found at https://revteccom.tk/downloads/RTC-FSEC.asc.

Click the “Import…” button on the top of the window, and select the key file you downloaded.



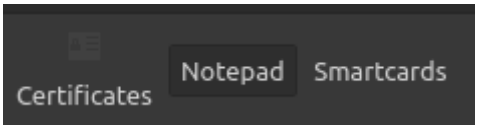
The “Import…” button here is the correct choice.

After the public key is imported, in the default “Certificates” screen, select the key you just imported, then right-click it and select “Change Certification Trust…” (on Linux) or “Change Certification Power” (on Windows), you will be faced with a range of trust options,

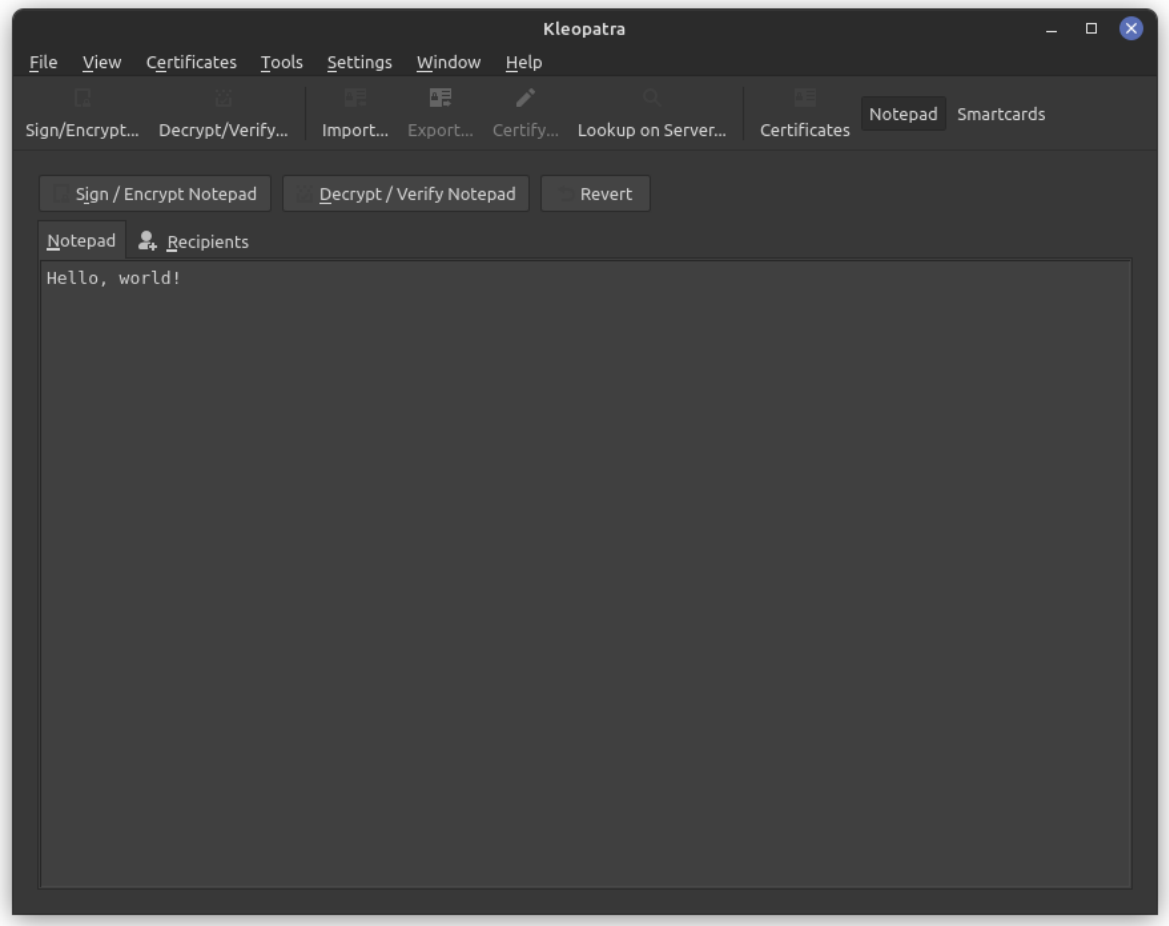
that’s up to you. Read the descriptions for each trust level carefully. For the RTC key, we recommend “Full Trust”.

Sending an encrypted message

Send an encrypted message by going to the “Notepad” screen, and typing in your message.



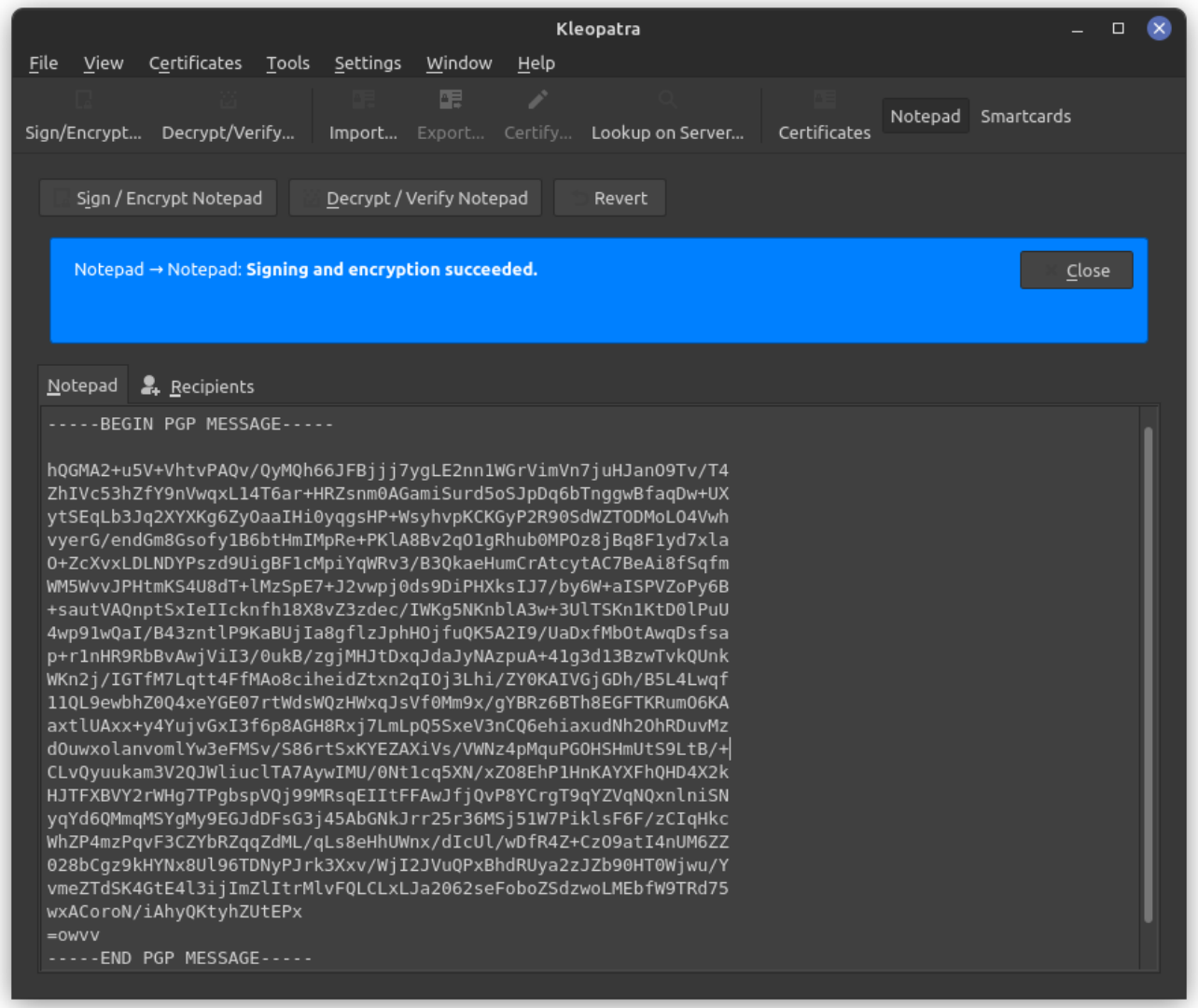
The “Notepad” section is where you can encrypt messages.



The notepad section, with “Hello, World!” written as the text body to be encrypted.

After writing our desired message, we move to the “recipients” section to configure who we will allow to decrypt the message, and if we will sign it, keeping in mind all previous remarks about anonymity. You may include as many receivers for the message as you want, by using the “Please enter a name or e-mail address…” field as a search box.

Then, we will move back to the “Notepad” section and press the “Sign/Encrypt Notepad” button. We will be presented with the garbled, encrypted version of our text.

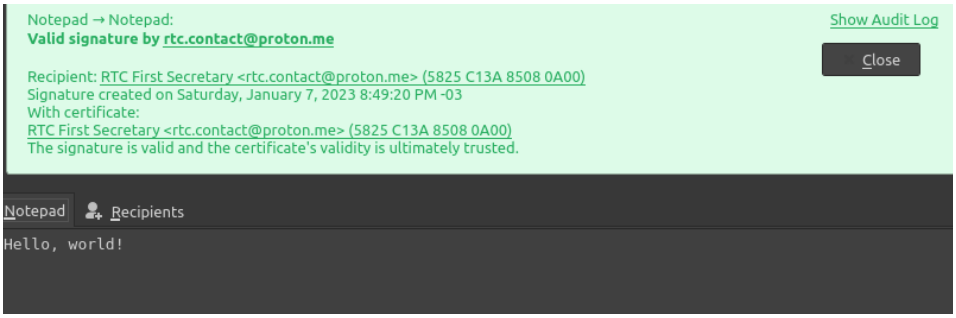


Then, we may use that content as our encrypted message, and paste it in an e-mail to send it over the internet to the specific person we addressed the message to. Send the whole text, including the begin and end indicators.

Receiving an encrypted message

When we receive an encrypted message, we may decrypt it by pasting it into the Notepad, with the begin and end blocks included, and clicking the “Decrypt/Verify Notepad” button. If the decryption is successful, you will see a success message, the

signature of the sender – including the date of signing – and the recipient of the message. In the case of the following example, the recipient is the signatory:



Letting people know

If you want to let people know of the existence of your public key, the most popular option is to post it to a keyserver, the most popular and trusted ones are:

- PGP Global Directory (https://keyserver.pgp.com/)
- keys.opengpg.org (https://keyserver.pgp.com/)
- Ubuntu PGP Keyserver (https://keyserver.ubuntu.com/)

Keep in mind **you should NEVER access keyservers trough HTTP**, **always use HTTPS**, as it encrypts your web requests, otherwise, your ISP can alter the content you see and download.

The safest option if you don't want to be too public with your public key is to upload it to a personal website, or upload it to a standard downloads site and link it on your social media profiles.

If your private key has been compromised and you wish to inform people of that, or you've stopped using your current key pair, you may upload what's called a "revocation certificate" to the places where you previously uploaded a public key, which is basically a version of your public key that informs the user that it's been revoked, so that when people look up and utilize the most recent version of your public key on a keyserver, it won't work.

You may generate a revocation certificate by right-clicking your key in the certificates menu, clicking the "Details" option and clicking "Generate revocation certificate".

Additional words on security

The encryption around your key and your messages is secure, and it will be until we're well into the era of quantum computing. Famously, nowadays, encryption isn't broken, instead, flaws that make it redundant are exploited. It doesn't matter how strong the encryption around your messages is if you're careless around your private keys.

Take into consideration the security guidelines and the remarks made in this text. An organization with a strong chain of trust and discipline can have a near military-grade system for exchanging messages right at their fingertips.