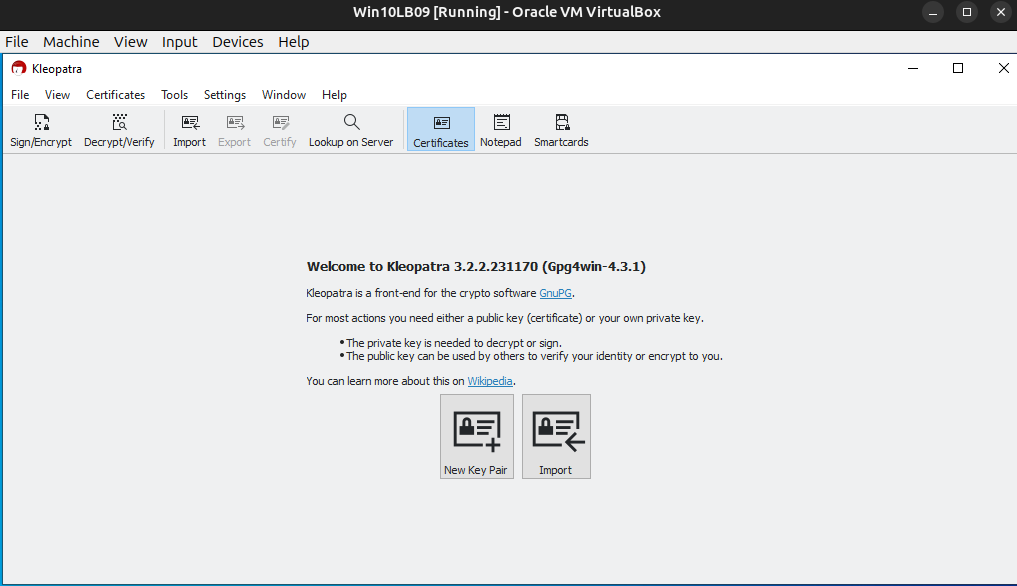
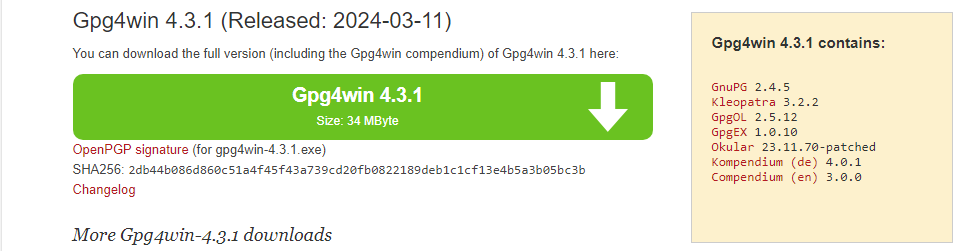
Hélio Ferreira 21/05/2024  
  
**Lab: Public Key Infrastructure (PKI)**

### **Part 1: Staging**

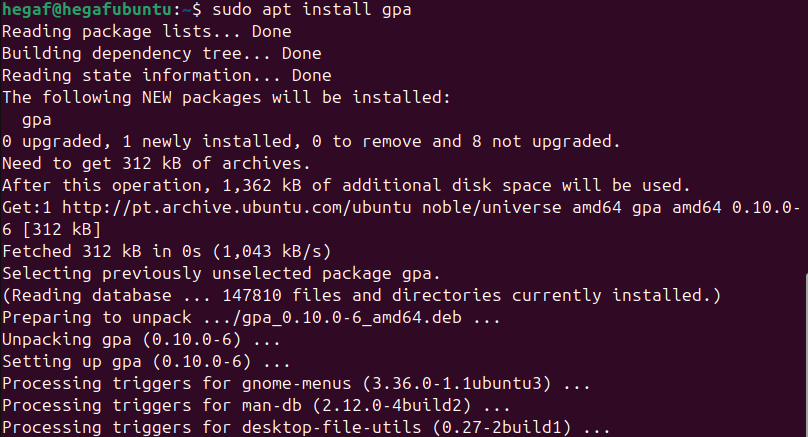
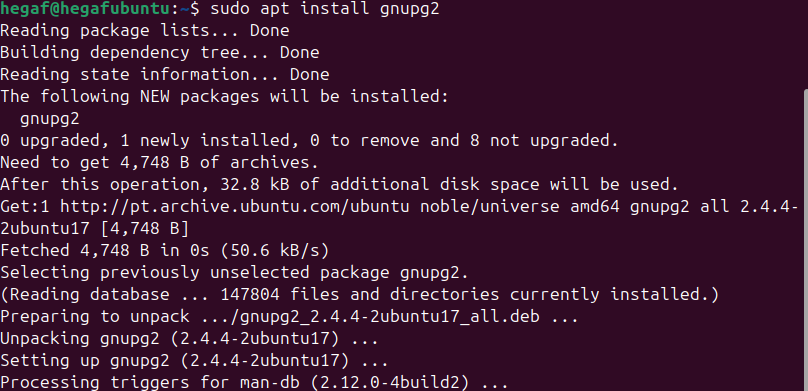
You will need two VMs for today’s lab:

* Windows 10
* Ubuntu Linux Desktop

First prepare the Windows 10 environment.

* Download and install [Gpg4win 3.1.13](https://www.gpg4win.org/).

Next prepare the Ubuntu Linux Desktop environment.

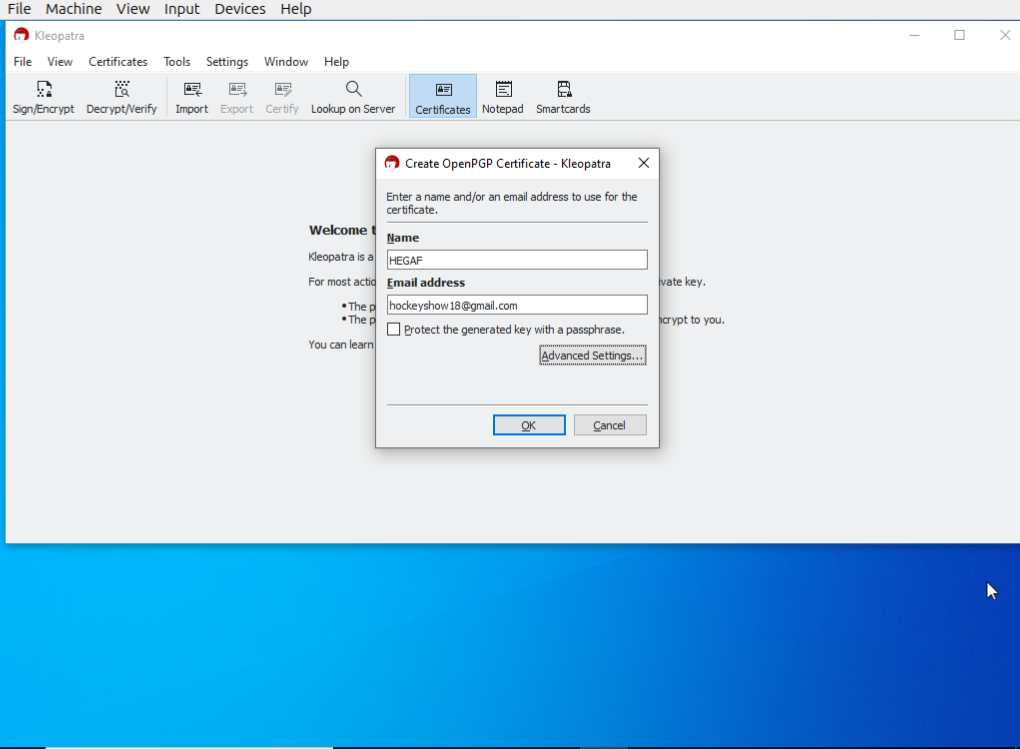
* Download and install **gnupg2** and **gpa**.

### **Part 2: Message Encryption with Gpg4win on Windows 10**

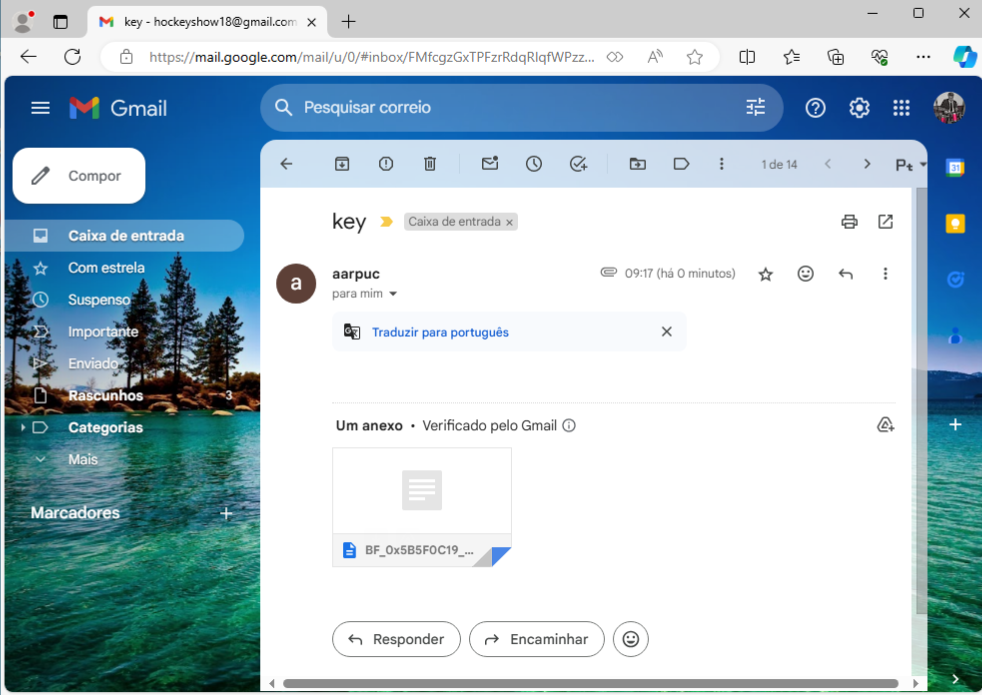
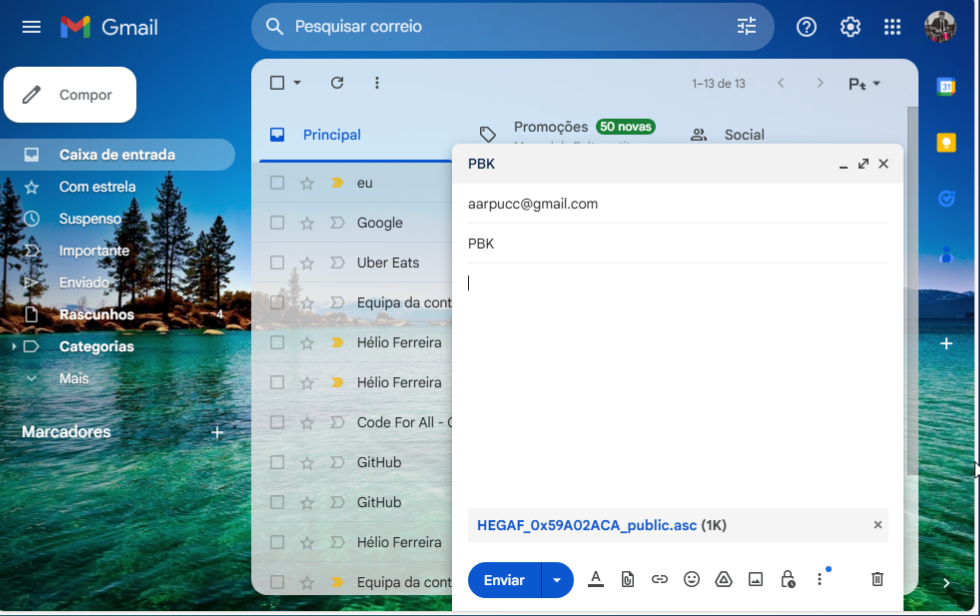
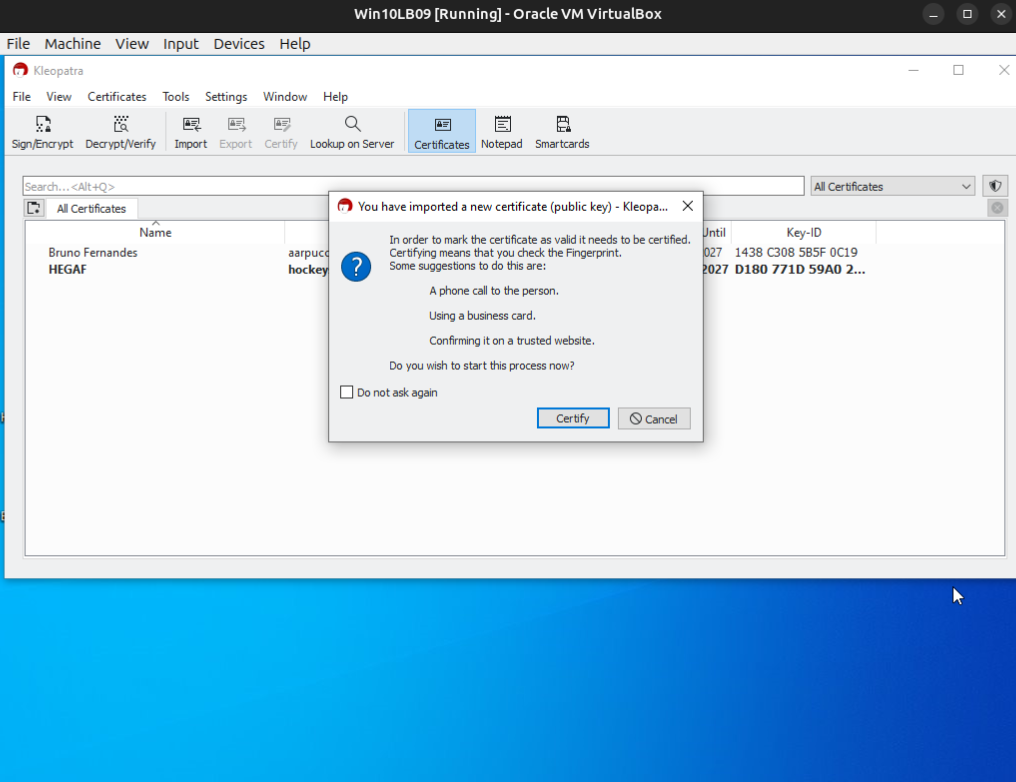
OpenPGP is usable in Windows environments as Gpg4win.

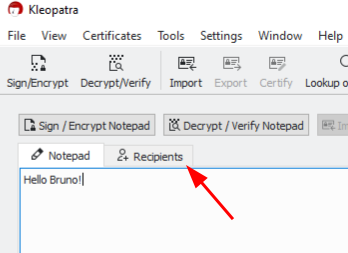
*You’ll need a partner for Part 2 and Part 3 of today’s lab. If you still don’t have a partner, contact your instructor.*

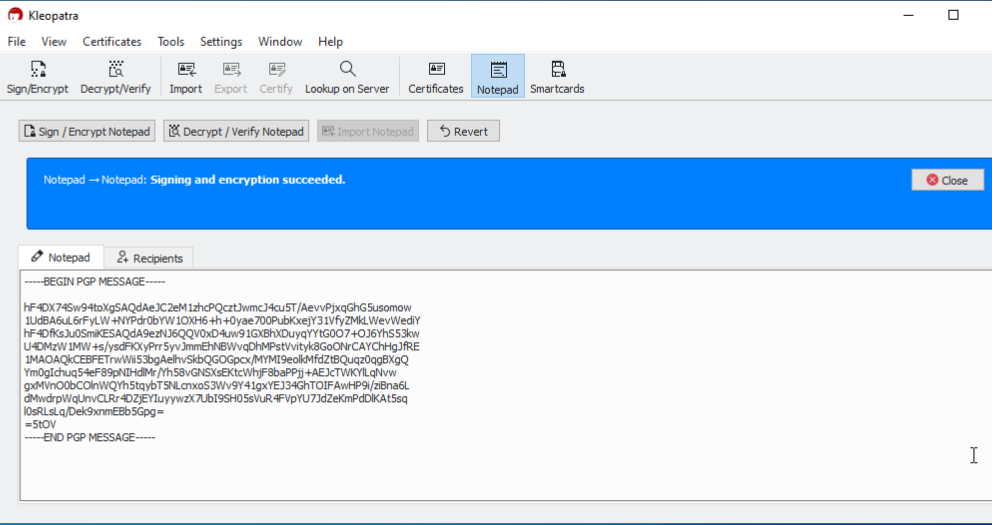
Each partner will perform the following operations on their own system:

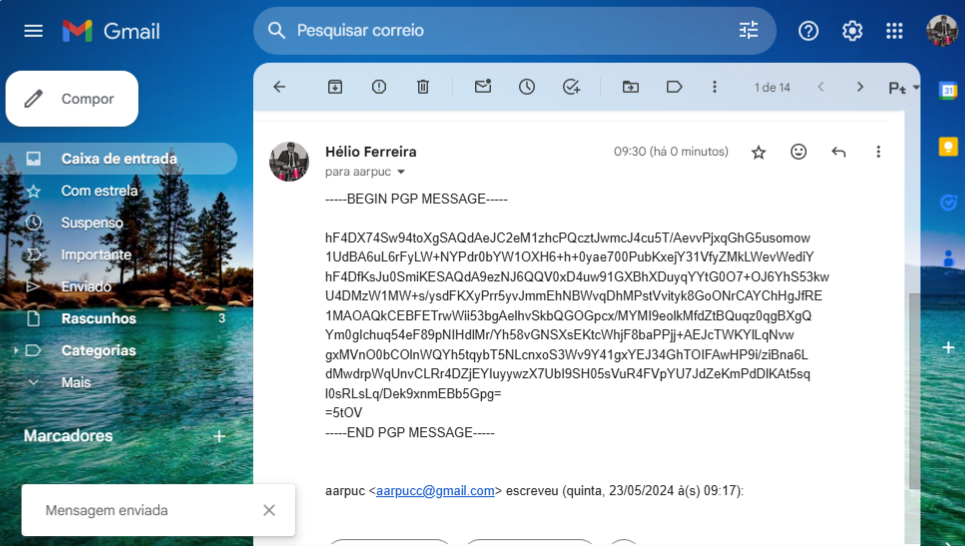
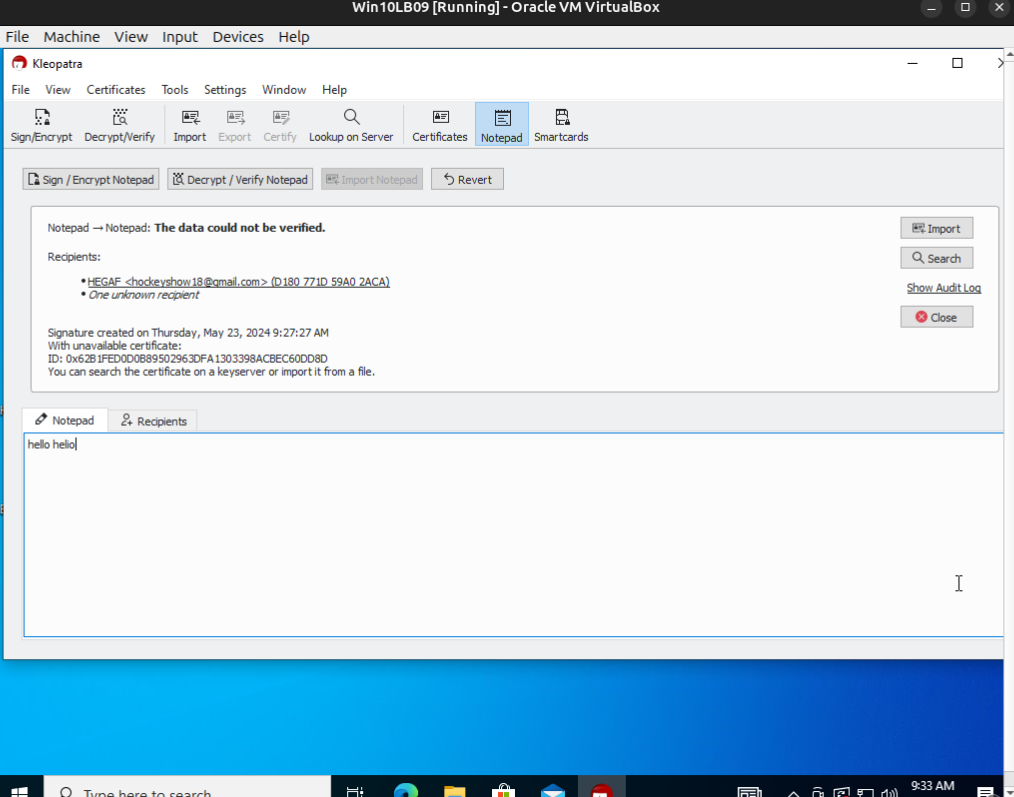
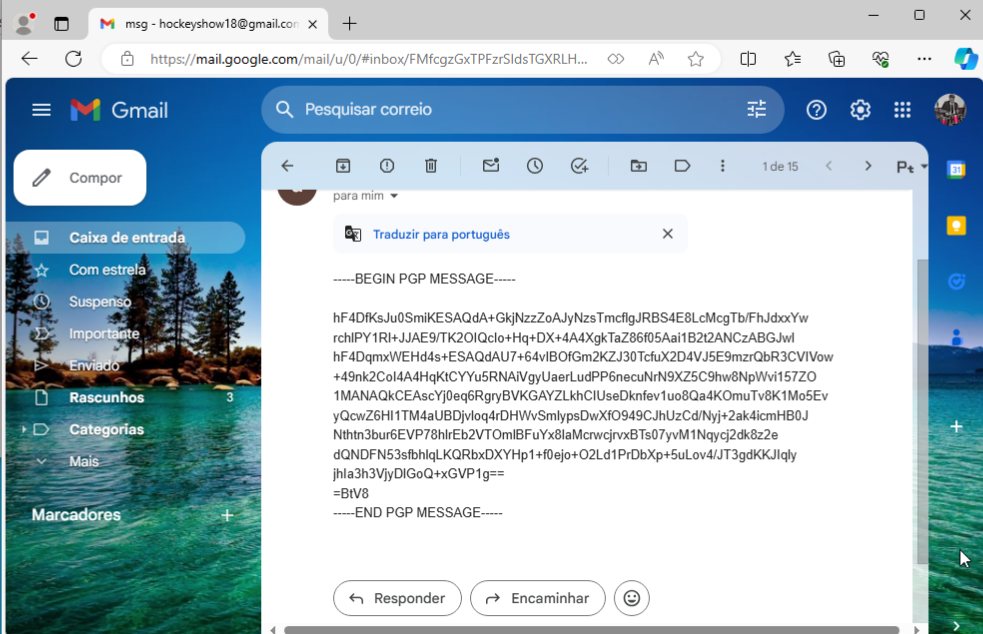
* Create a key pair in Kleopatra.
* Export the public key to your desktop as an OpenPGP Certificate (include your name in the file name to differentiate between your public key and your partner’s) and view it with a text editor.



* Send this public key to your partner and receive theirs.
* Load your partner’s key into Kleopatra and associate it with their name.
* Use the “Notepad” feature in Kleopatra to encrypt a secret message with your partner as the recipient

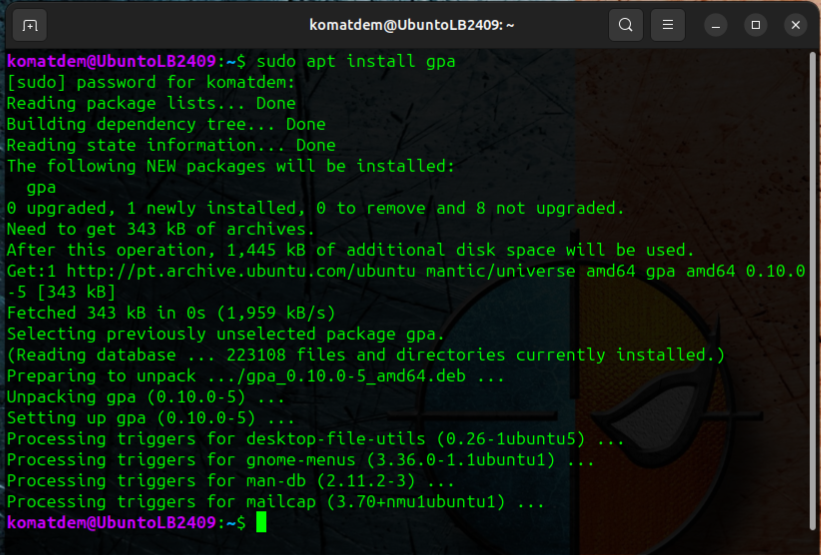
.

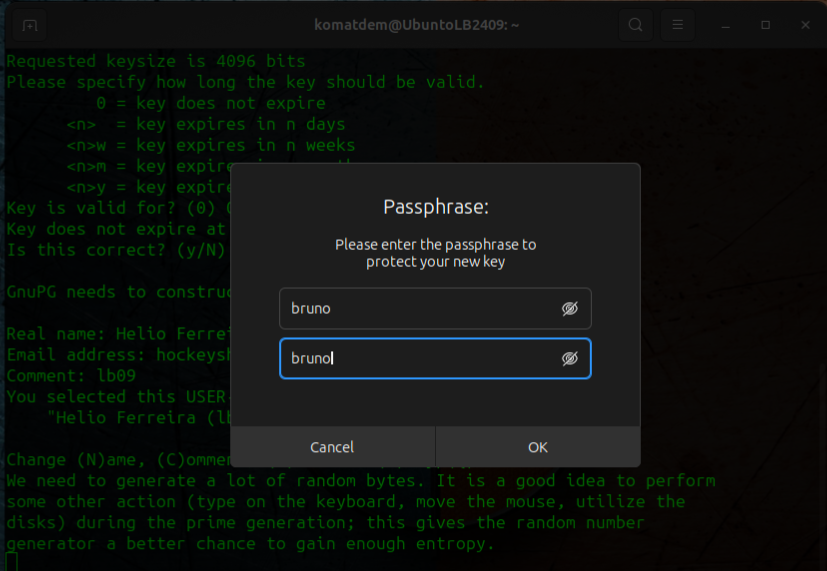
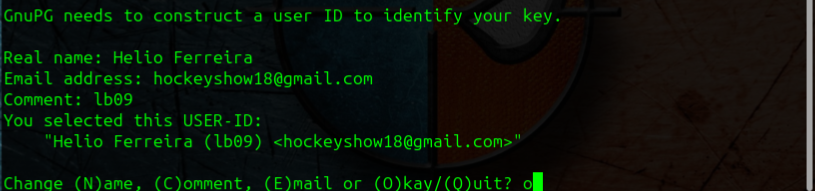
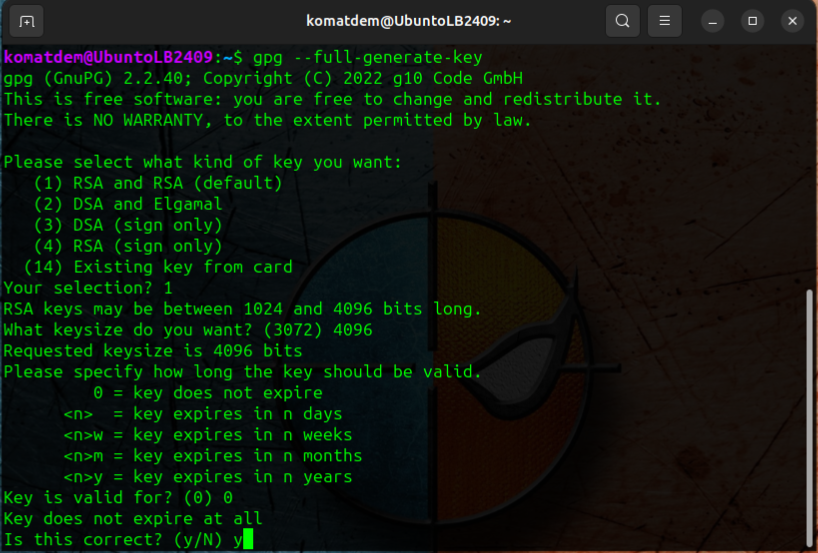
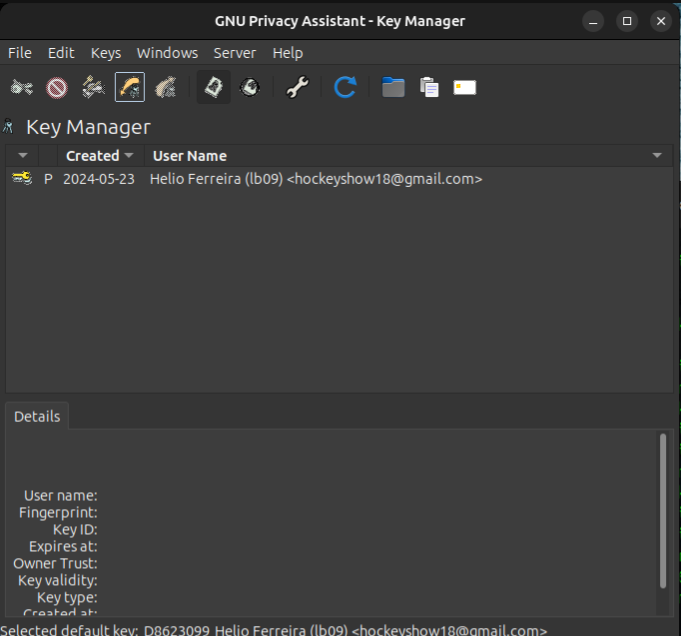


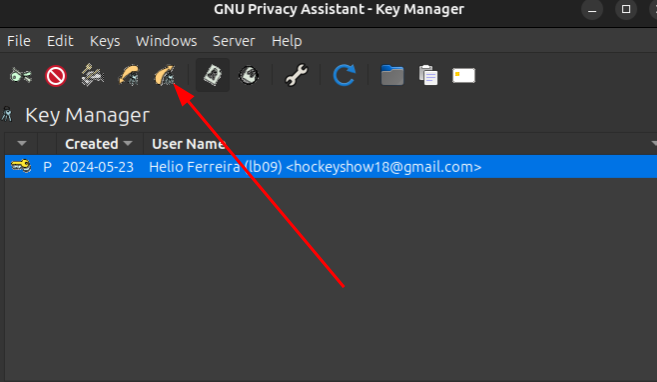
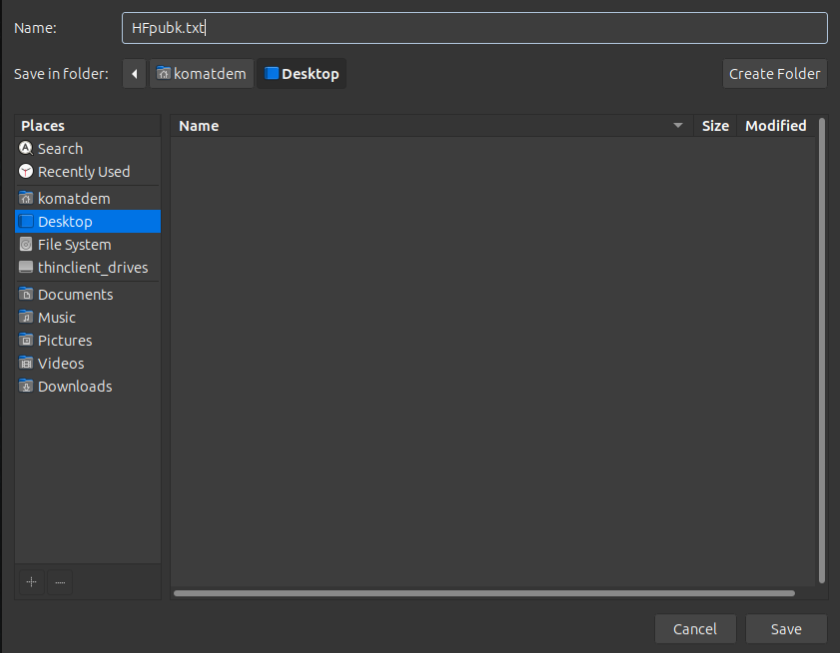
* Copy the complete message text (including the complete header and footer designating the beginning and end of the key block) into an email and send it to your partner.
* Receive their message, copy it into the “Notepad” feature, and use Kleopatra to decrypt the message using your private key.

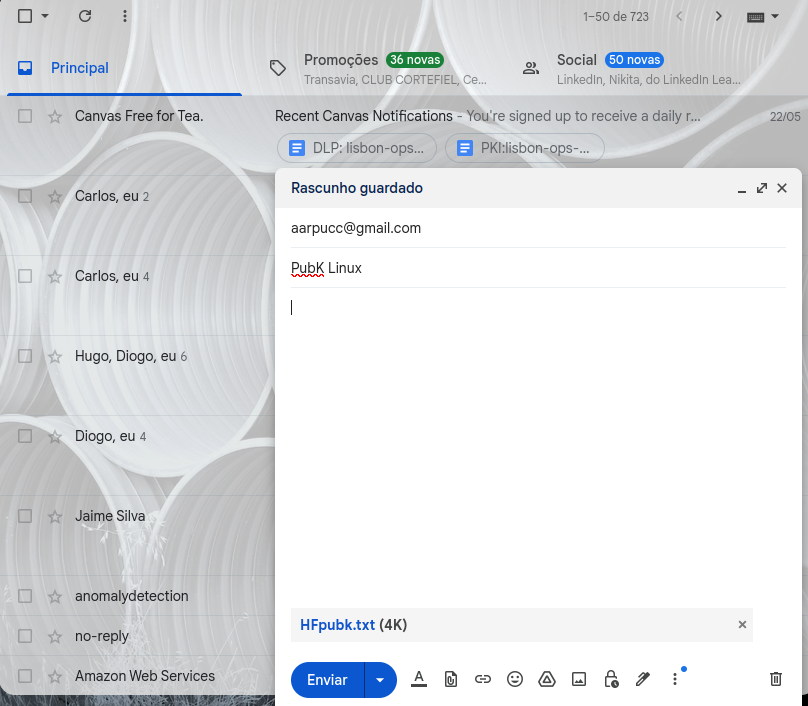
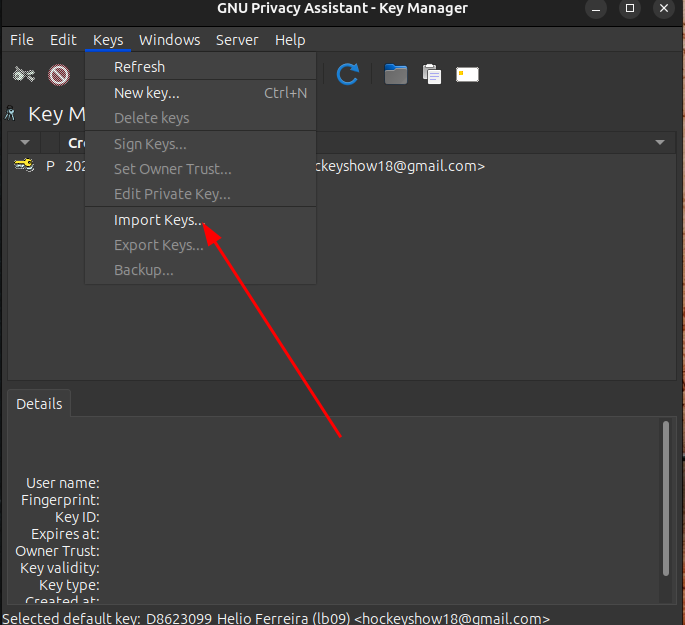
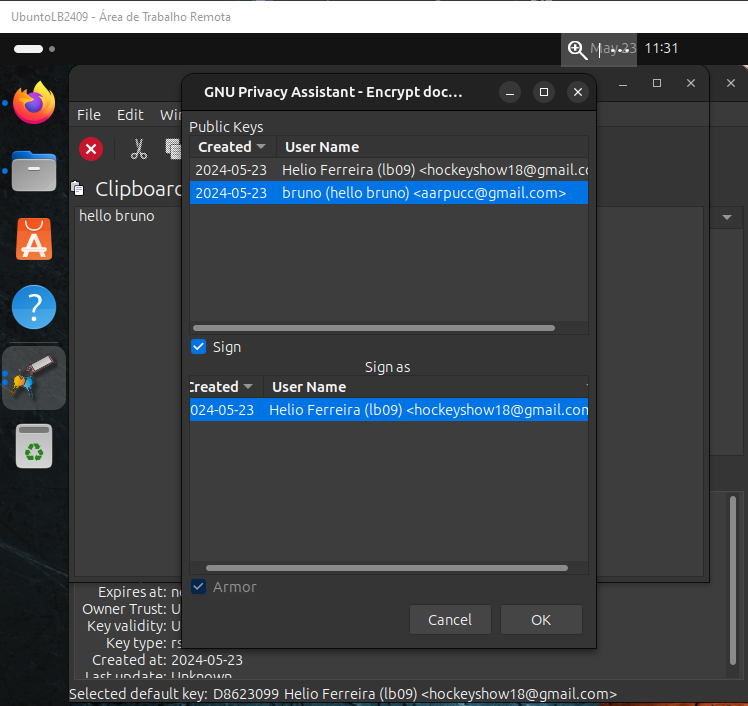
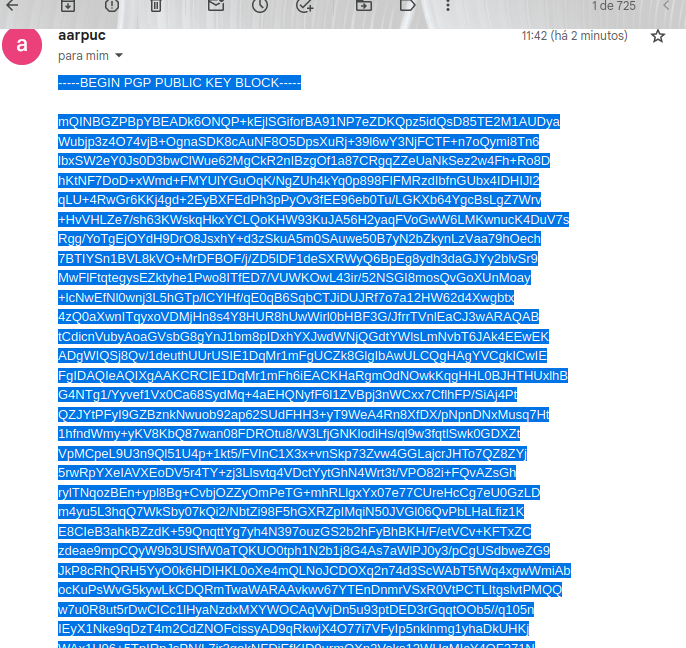
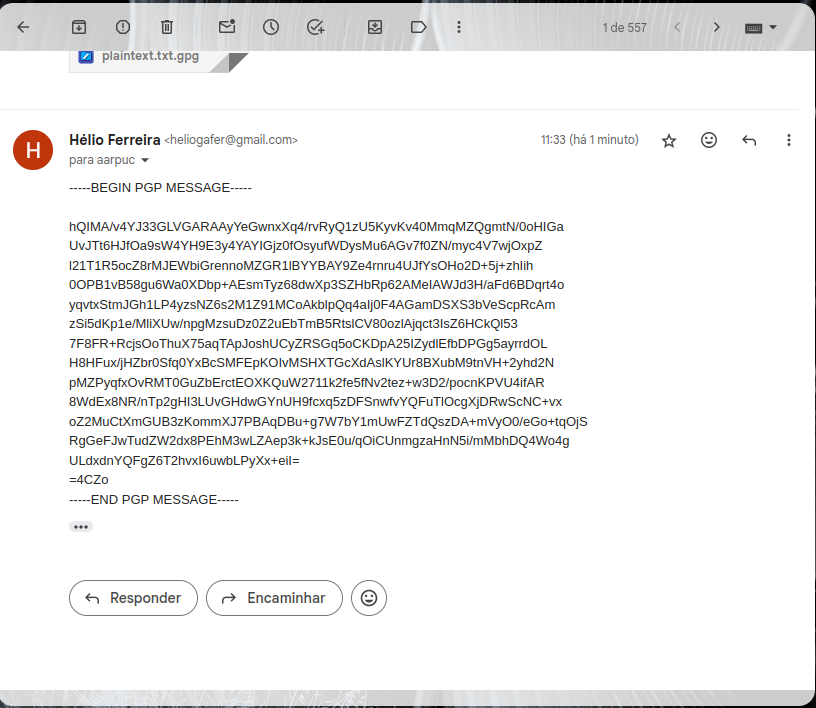
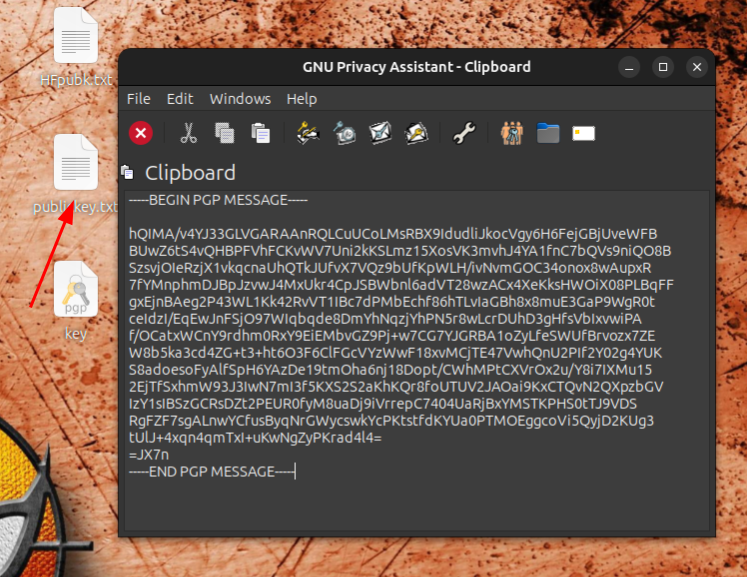
Once you have successfully decrypted your partner’s message into cleartext, this part of today’s lab is complete. Include in your lab submission a screenshot of the ciphertext and cleartext.

### **Part 3: Email Encryption with GPG on Ubuntu Linux Desktop**

OpenPGP is usable in Linux environments as GNU Privacy Guard (GPG). Each partner will perform the following operations on their own system:

* Perform a full key pair generation operation in GPG:
  + RSA and RSA
  + 4096 key size
  + Key should not expire
* Launch GNU Privacy Assistant (GPA) GUI key manager application. Your new key pair from GPG should appear here.
* Export your public key to your Ubuntu Linux desktop with a .txt file extension as “public\_key”.



* Share your public key with your lab partner. If you like, [publish your public key](https://security.stackexchange.com/questions/406/how-should-i-distribute-my-public-key) or through a service like keybase.io.
* Import your partner’s public key into GPA.
* Compose a cleartext secret message.  
    
  The encrypt message was “hello bruno”.
* Encrypt the secret message with your recipient’s public key and sign it.
* Send the encrypted message to your lab partner.
* When you receive the encrypted message from your lab partner, decrypt it.

Once you have successfully decrypted your partner’s message into cleartext, this part of today’s lab is complete. Include in your lab submission a screenshot of the ciphertext and cleartext.

### **Part 5: Reporting**

Answer the below discussion prompts in your submission today.

* How can public key encryption utilities ensure integrity and confidentiality of data between two parties?  
    
  Public key encryption secures data privacy (confidentiality) between two parties. They create key pairs: a public key for sharing and a private key for secrecy. Data gets encrypted with the receiver's public key, making it unreadable without their private key. Digital signatures, often used alongside, ensure data integrity by verifying the sender and preventing tampering. Popular tools include GnuPG and OpenPGP.
* Is PGP a secure protocol to be using?  
    
  PGP can be secure, but it's not perfect.

**Pros:** Trusted, strong encryption, open standard.

**Cons:** Complex for users, limited adoption, vulnerable to social engineering.

**Alternatives:** End-to-end encrypted messaging apps (might be easier to use, but security depends on the app).

**Overall:** Good for tech-savvy users, complex for others. Consider your needs and expertise before using PGP.

* Why is PGP not very popular among casual computer users?  
    
  PGP is too complex for casual users. It requires:

**Learning cryptography basics** (key pairs, encryption/decryption)

**Managing multiple keys and passwords** securely

**Installing extra software** for PGP functionality

**Finding contacts who also use PGP**

Casual users prefer:

**Simpler, built-in encryption** in messaging apps

**Convenience and familiarity** over technical control

Identify a website on the internet that does NOT have HTTPS (SSL) encryption in place.  
  
**http://info.cern.ch**

* Why do you think the website lacks SSL encryption?  
    
  There are two reason <http://info.cern.ch> lacks SSL encryption are:  
    
  1 - It was the first internet site, so, it is too old tu use SSL and,  
  2 - It is a informational website that doesn't require sensitive data from users. Websites that handle sensitive data like passwords or credit cards typically use HTTPS for encryption.
* What steps should the web administrator take in order to reconfigure the site to use SSL encryption?  
    
  The steps a web administrator takes to reconfigure a website for SSL encryption can vary depending on the specific hosting provider and server setup. However, here's a general guideline:

1. **Obtain an SSL Certificate:** There are different types of SSL certificates available, depending on the website's needs and validation level. The administrator can choose a free or paid certificate from a certificate authority (CA).
2. **Generate Server Configuration:** Using the chosen SSL certificate, the administrator will need to generate server configuration files. This might involve creating a Certificate Signing Request (CSR) and installing the certificate and private key on the web server.
3. **Configure the Web Server:** The web server software needs to be configured to use the SSL certificate and private key. This typically involves editing configuration files specific to the web server software (e.g., Apache, Nginx).
4. **Enable HTTPS Redirects:** To ensure all traffic goes through the secure connection, the administrator will need to configure the server to automatically redirect any HTTP requests (http://) to HTTPS requests (https://).
5. **Test the Configuration:** After implementing the changes, the administrator should thoroughly test the website to ensure everything functions correctly under HTTPS. This includes checking for broken links, compatibility issues, and a valid SSL certificate on the website.