

Міністерство освіти і науки України  
Національний авіаційний університет  
Факультет кібербезпеки, комп'ютерної та програмної інженерії  
Кафедра комп'ютеризованих систем управління

Лабораторна робота № 2.4  
з дисципліни «Захист інформації в комп'ютерних системах»  
на тему «Криптографічні програмні засоби з відкритим ключем»

Виконав:  
студент ФККПІ  
групи СП-425  
Клокун В. Д.  
Перевірила:  
Супрун О. М.

Київ 2019

## 1. МЕТА РОБОТИ

Ознайомитися з основними поняттями асиметричних криптографічних систем.

## 2. ЗАВДАННЯ РОБОТИ

Встановити програмне забезпечення для ОС Windows та GNU/Linux, що виконує криптографічні операції з відкритим ключем, та навчитися його використовувати.

## 3. ХІД РОБОТИ

### 3.1. Асиметричне шифрування в операційній системі Windows

Щоб виконати лабораторну роботу, готуємо систему під управлінням операційної системи Windows і встановлюємо на неї програмний продукт PGP Desktop. Встановивши його, генеруємо публічний та приватний ключі для асиметричного шифрування (рис. 1).

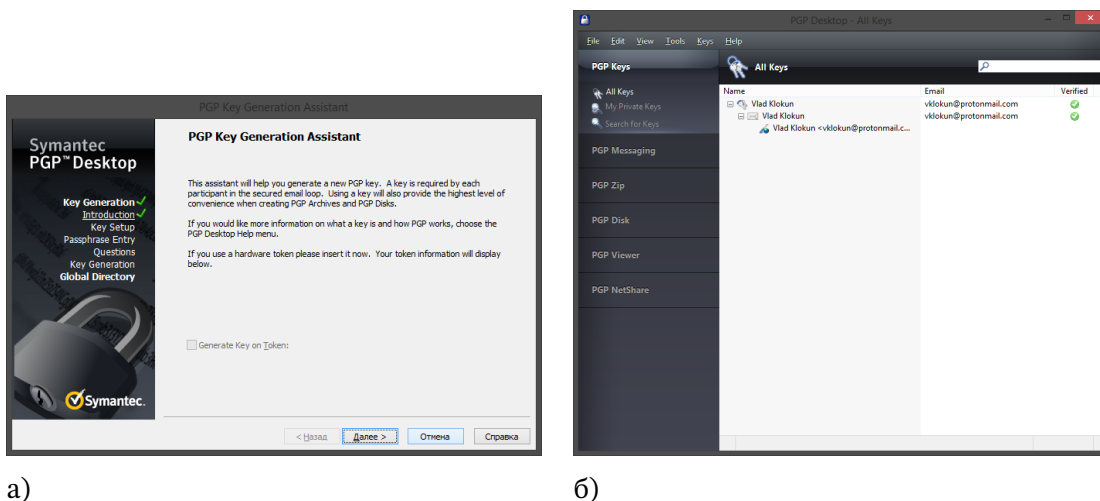


Рис. 1: Створення пари ключів: публічного і приватного

Експортуємо створений публічний ключ (рис. 2). Для цього обираємо створений ключ, викликаємо контекстне меню, обираємо пункт **Export...** і зберігаємо у бажаній директорії.

Переглядаємо список ключів, до яких має доступ наша криптографічна система з відкритим ключем (КСВК) (рис. 3).

Знаходимо відкриті ключі інших осіб та завантажуюмо будь який з них, а також перевіряємо його цілісність за наведеним відбитком і імпортуємо

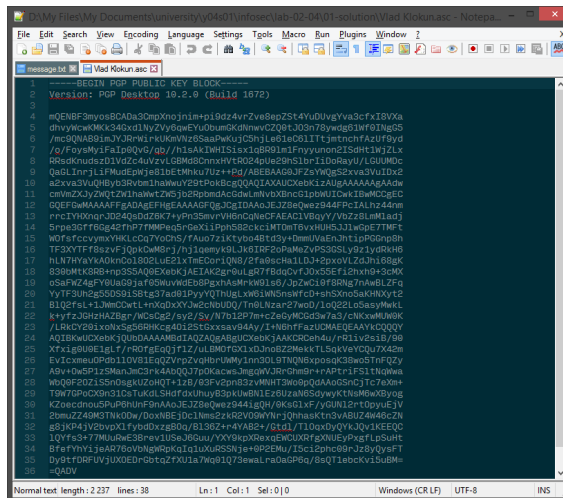


Рис. 2: Результат експорту створеного публічного ключа

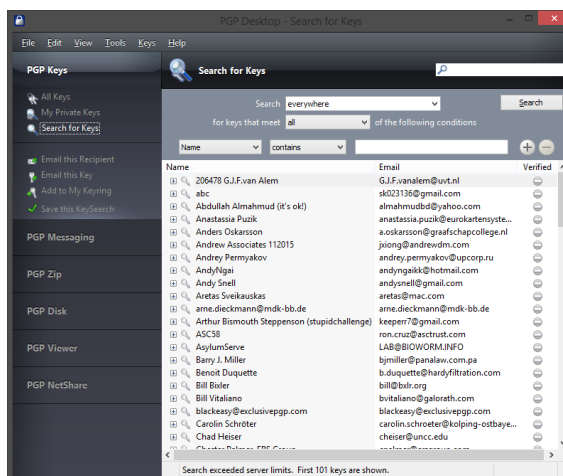


Рис. 3: Список ключів, доступних системі

його (рис. 4).

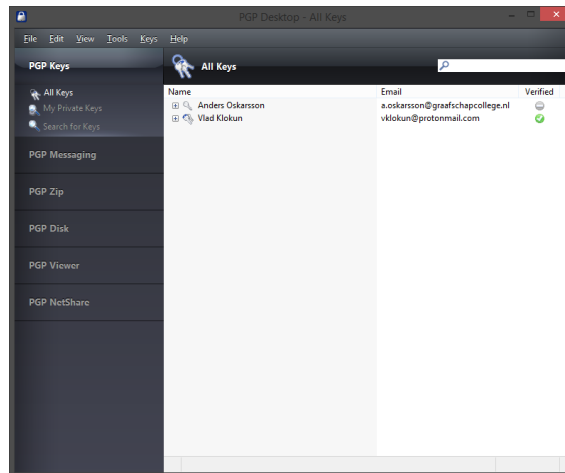


Рис. 4: Результат імпорту ключа

Шифруємо просте повідомлення за допомогою власного публічного ключа. Отримавши результат шифрування, розшифровуємо його і перевіряємо дані (рис. 5).

Підписуємо файл цифровим підписом та передаємо файл і цифровий підпис товаришу. Отримуємо аналогічну пару від товариша і перевіряємо справжність файлу (рис. 6).

Підписуємо публічний ключ іншої людини, знайдений серед ключів, доступних криптографічній системі (рис. 7).

Додаємо і видаляємо компоненти ключа. Для цього заходимо у властивості створеного ключа і генеруємо новий підключ.

Створюємо відкликаючий сертифікат та відкликаємо ключ з його допомогою (рис. 9).

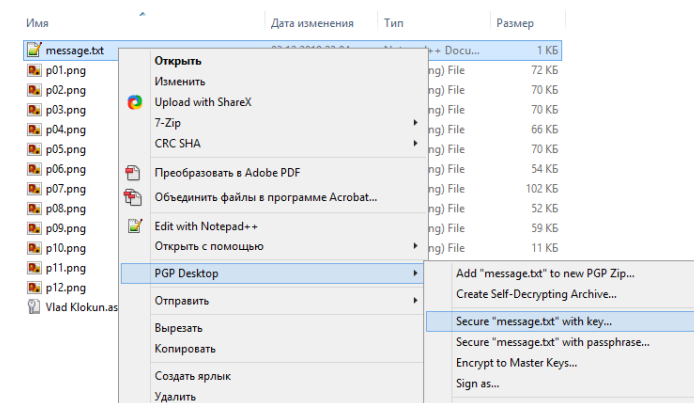
### **3.2. Асиметричне шифрування в операційній системі GNU/Linux**

Для роботи з криптографією з відкритим ключем в операційній системі GNU/Linux будемо використовувати програмну систему GNU Privacy Guard. Генеруємо пару ключів: публічний і приватний (рис. 10).

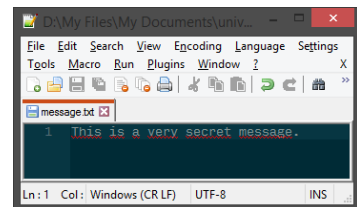
Експортуємо згенерований ключ і перевіряємо список усіх доступних ключів (рис. 11).

Знаходимо ключ іншої людини, імпортуємо та перевіряємо його цілісність (рис. 12).

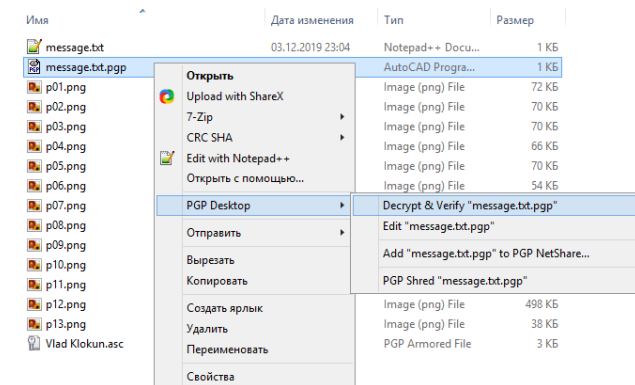
Шифруємо та дешифруємо просте повідомлення за допомогою публічного ключа. Потім підписуємо повідомлення і перевіряємо його цілісність (рис. 13).



а)

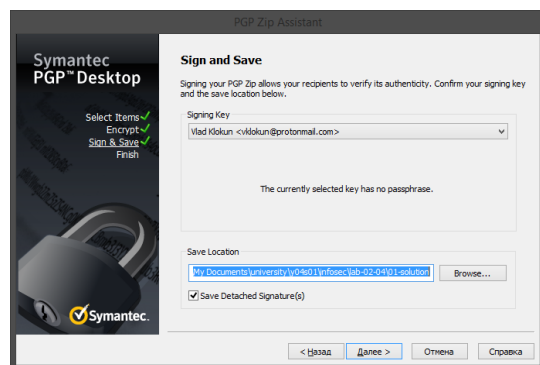


б)

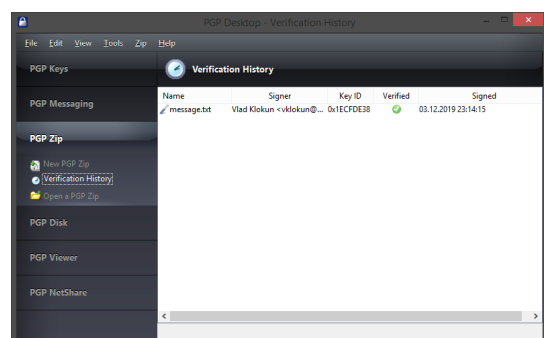


в)

Рис. 5: Шифрування, розшифрування і результат

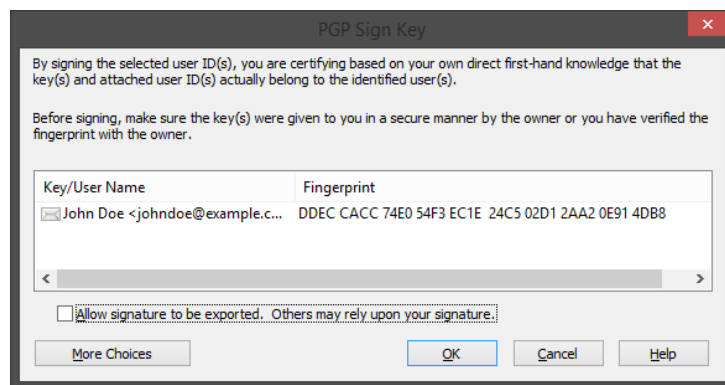


а)



б)

Рис. 6: Результаты цифрового подпису та перевірки підписаного файлу

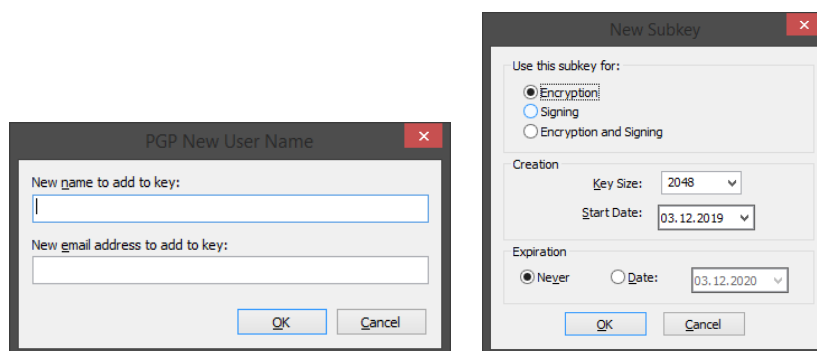


a)

Name	Email	Verified
Anders Oskarsson	a.oskarsson@graafschapcollege.nl	
John Doe	johndoe@example.com	
Vlad Klokun	vklokun@protonmail.com	

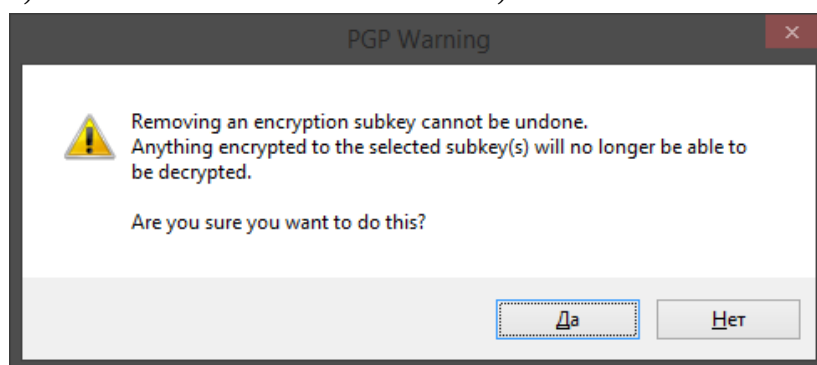
б)

Рис. 7: Результати цифрового підпису та перевірки підписаного публічного ключа



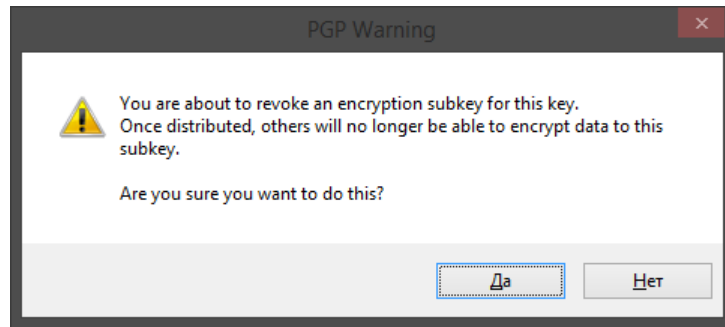
a)

б)

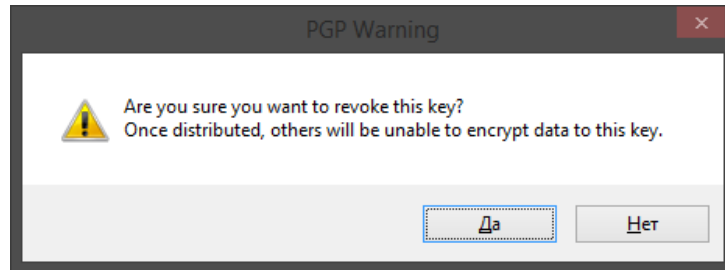


в)

Рис. 8: Додавання і видалення компонентів ключа



а)



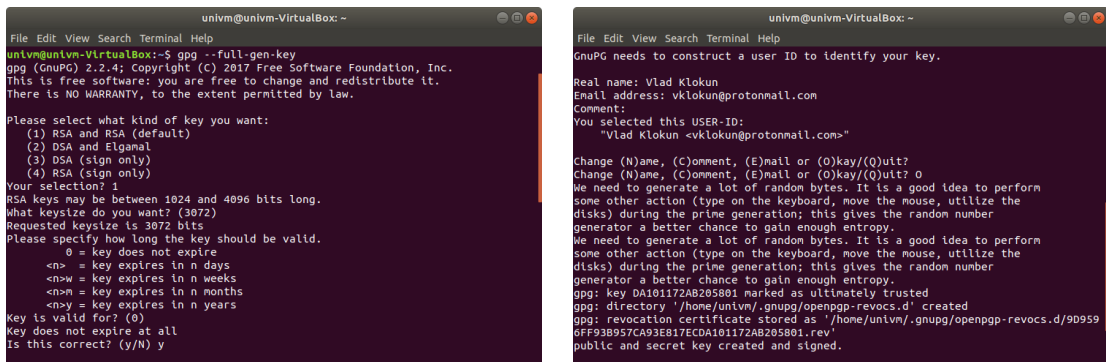
б)

Рис. 9: Додавання і видалення компонентів ключа

Отже, на цьому робота з криптографічною системою з відкритим ключем завершена.

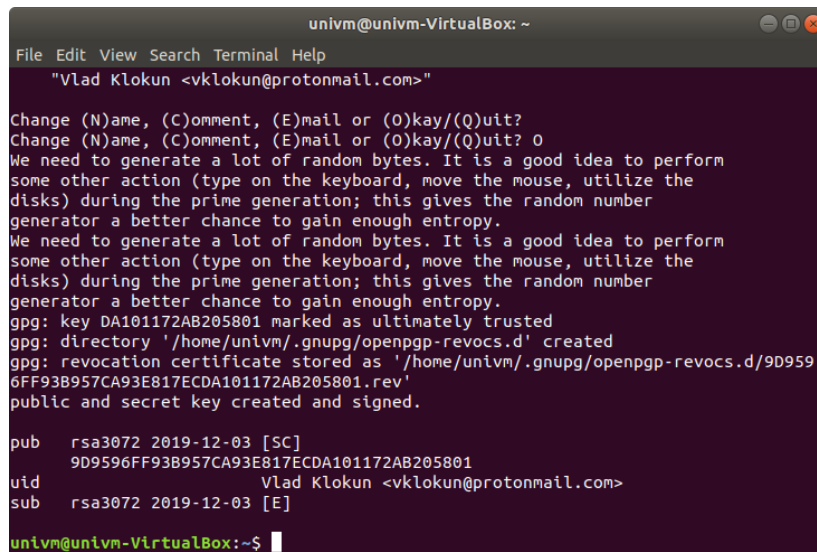
#### 4. Висновок

Виконуючи дану лабораторну роботу, ми ознайомилися з основними поняттями асиметричних криптографічних систем і встановили програмне забезпечення для ОС Windows та GNU/Linux, що виконує криптографічні операції з відкритим ключем, та навчилися його використовувати.



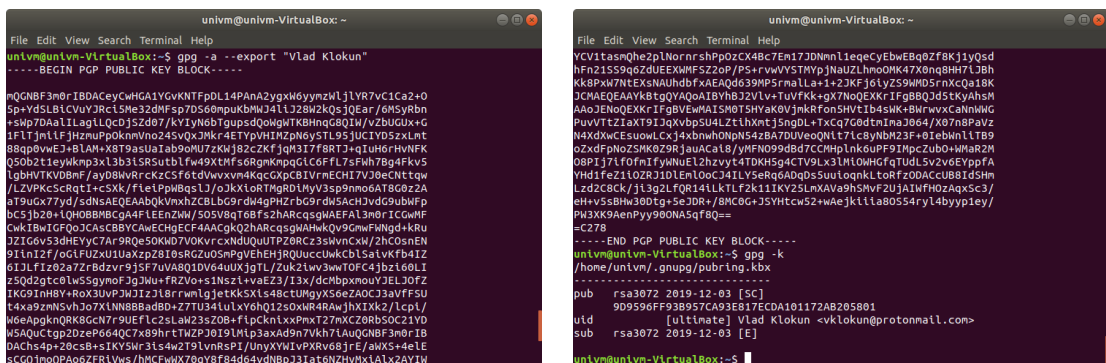
a)

b)



b)

Рис. 10: Генерація публічного і приватного ключа в GNU Privacy Guard

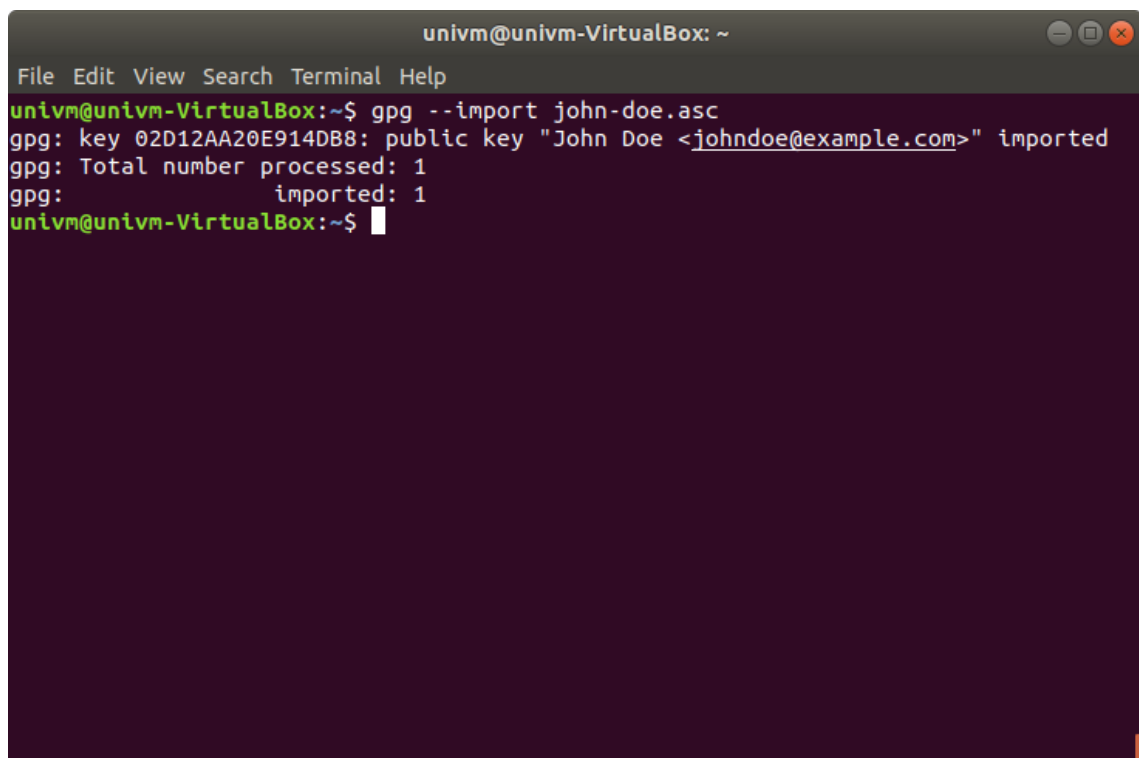


a)

b)

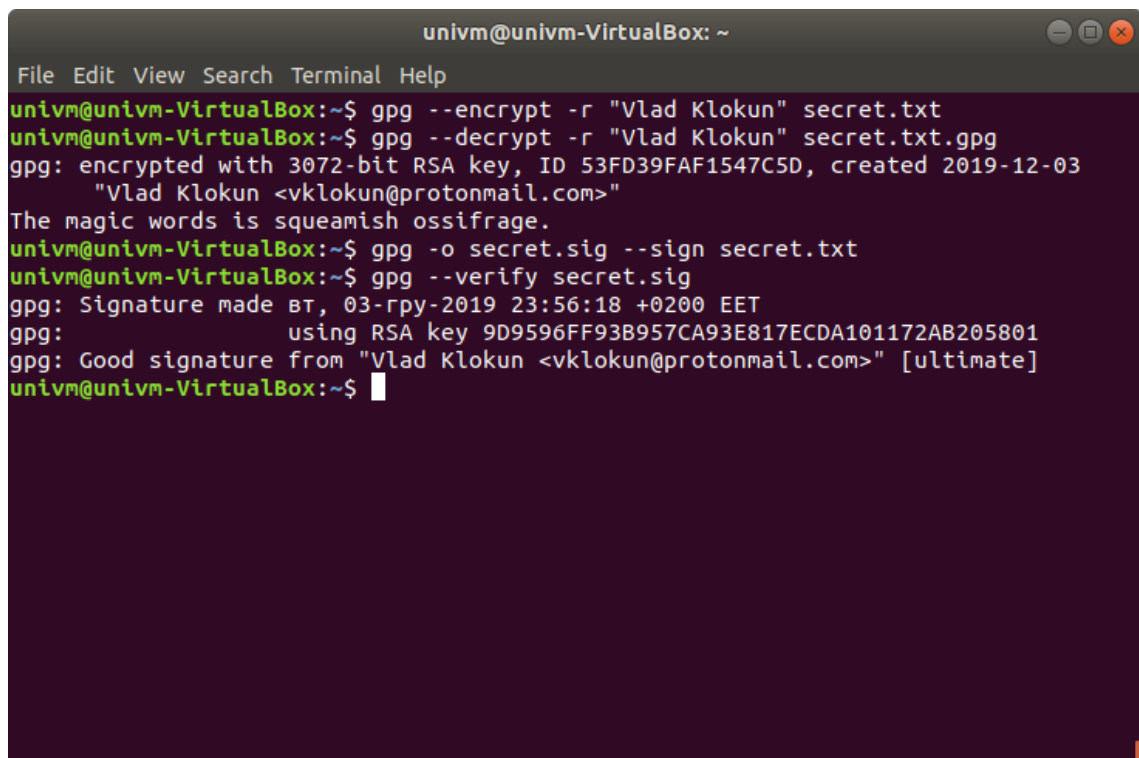
Рис. 11: Експорт згенерованого ключа і перевірка списку доступних ключів в GNU Privacy Guard





```
univm@univm-VirtualBox: ~  
File Edit View Search Terminal Help  
univm@univm-VirtualBox:~$ gpg --import john-doe.asc  
gpg: key 02D12AA20E914DB8: public key "John Doe <johndoe@example.com>" imported  
gpg: Total number processed: 1  
gpg:             imported: 1  
univm@univm-VirtualBox:~$
```

Рис. 12: Імпорт і перевірка ключа іншої людини в GNU Privacy Guard



```
univm@univm-VirtualBox: ~  
File Edit View Search Terminal Help  
univm@univm-VirtualBox:~$ gpg --encrypt -r "Vlad Klokun" secret.txt  
univm@univm-VirtualBox:~$ gpg --decrypt -r "Vlad Klokun" secret.txt.gpg  
gpg: encrypted with 3072-bit RSA key, ID 53FD39FAF1547C5D, created 2019-12-03  
"Vlad Klokun <vklokun@protonmail.com>"  
The magic words is squeamish ossifrage.  
univm@univm-VirtualBox:~$ gpg -o secret.sig --sign secret.txt  
univm@univm-VirtualBox:~$ gpg --verify secret.sig  
gpg: Signature made Вт, 03-гpy-2019 23:56:18 +0200 EET  
gpg: using RSA key 9D9596FF93B957CA93E817ECDA101172AB205801  
gpg: Good signature from "Vlad Klokun <vklokun@protonmail.com>" [ultimate]  
univm@univm-VirtualBox:~$
```

Рис. 13: Шифрування, розшифрування, підпис і перевірка цілісності повідомлення в GNU Privacy Guard