

Лабораторная работа №3.
Программа для шифрования и подписи GPG,
пакет Gpg4win

Евсеев Дмитрий

20 июня 2016 г.

Оглавление

1	Цель работы	2
2	Описание работы	2
3	Ход работы	2
	3.1 Создание ключевой пары gpg	2
	3.2 Экспорт сертификата	4
	3.3 Постановка ЦП на файл	4
	3.4 Импорт сертификата и его подпись	7
	3.5 Расшифровка файла	8
4	Использование GNU Privacy handbook	9
5	Вывод	11

1 Цель работы

Научиться создавать сертификаты, шифровать файлы и ставить ЭЦП.

2 Описание работы

Электронная подпись (ЭП) – это особый реквизит документа, который позволяет установить отсутствие искажения информации в электронном документе с момента формирования ЭП и подтвердить принадлежность ЭП владельцу. Значение реквизита получается в результате криптографического преобразования информации.

3 Ход работы

Работа на данном этапе производится во frontend для gpg kleopatra. Kleopatra — инструмент для управления сертификатами X.509 и ключами gpg.

3.1 Создание ключевой пары gpg

Во вкладке *"File"* выбираем *"New certificate"*. В открывшемся окне вводим информацию: имя ключа, адрес почты, комментарии (рисунок 1).

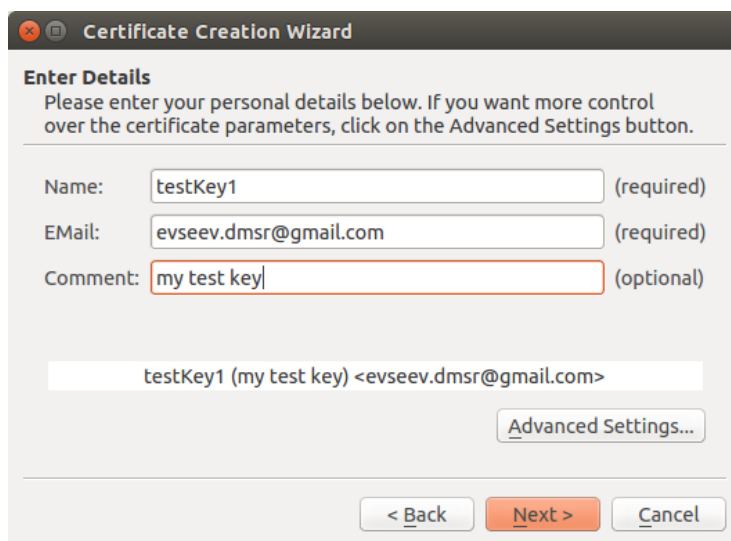


Рис. 1: Окно для ввода персональных данных.

Далее открывается окно подтверждения, в котором можно еще раз проверить информацию (рисунок 2)

Далее система просит ввести фразу-пароль. Появляется предупреждение о том, что программе нужно генерировать большое множество псевдослучайных величин и для этого не плохо было бы вести активную работу (перемещать мышь, печатать и т. д.), увеличивая тем самым энтропию (рисунок 3)

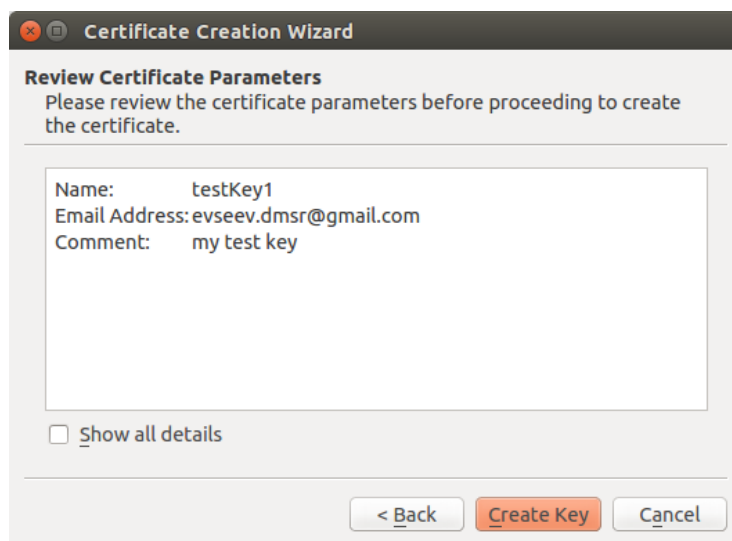


Рис. 2: Окно подтверждения.

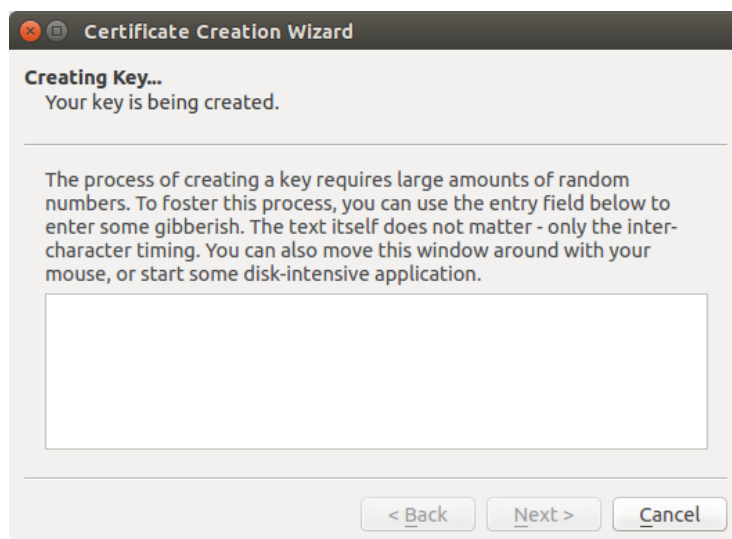


Рис. 3: Информационное окно.

Теперь можно увидеть новый сертификат в списке всех сертификатов (рисунок 4)

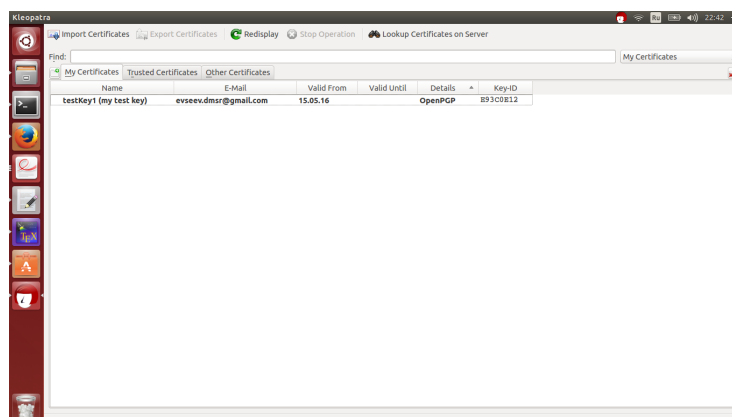


Рис. 4: Список сертификатов.

3.2 Экспорт сертификата

Для экспорта сертификата во вкладке *"File"* выбираем *"Export Certificate"*. После чего введем имя файла *testKey.asc* (рисунок 5).

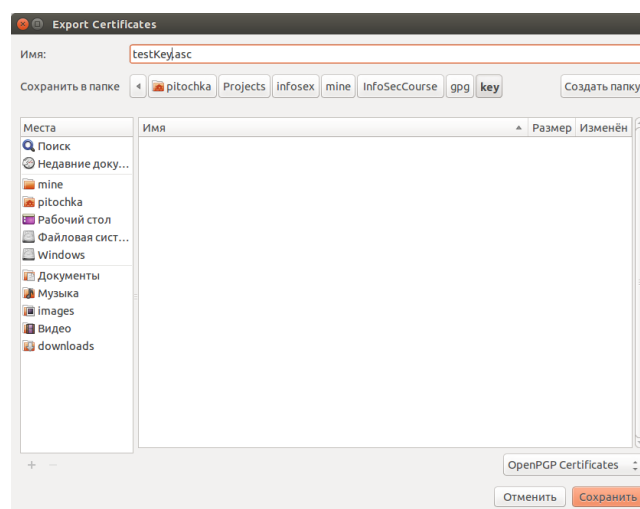


Рис. 5: Экспорт сертификата.

3.3 Постановка ЦП на файл

Для того, что бы поставить ЦП, во вкладке *"File"* выбираем *"Sign/Encrypt Files"* и выберем файл, на который необходимо поставить ЭЦП. В нашем случае это *readme.txt* (рисунок 6).

После выберем одно из трех предложенных действий.

- Sign and Encrypt
- Encrypt

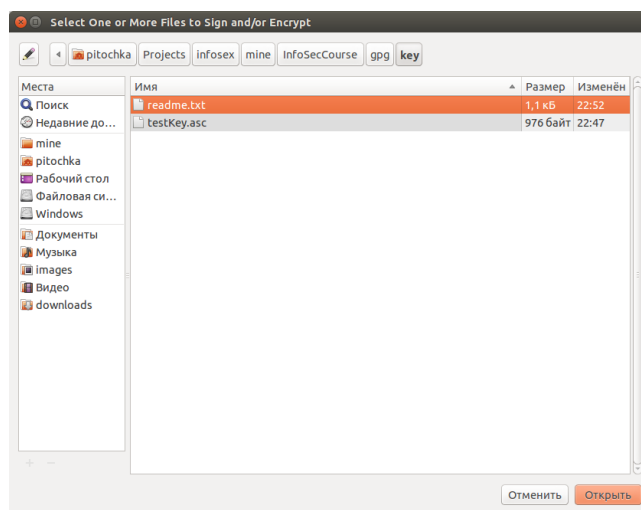


Рис. 6: Установка ЦП.

- Sign

В нашем случае *Sign* - создание цифровой подписи (рисунок 7)

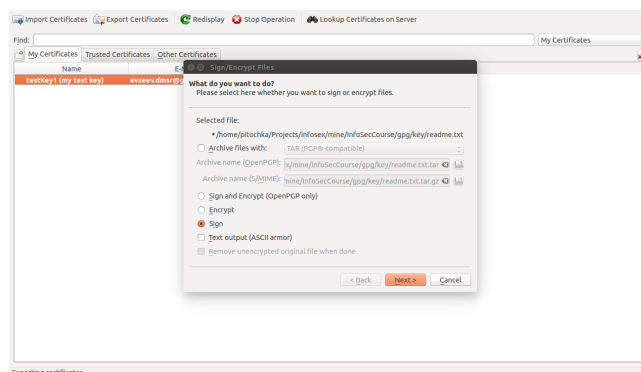


Рис. 7: Выбор стандарта и сертификата.

В открывшемся окне выбираем стандарт и один из имеющихся сертификатов(рисунок 8).

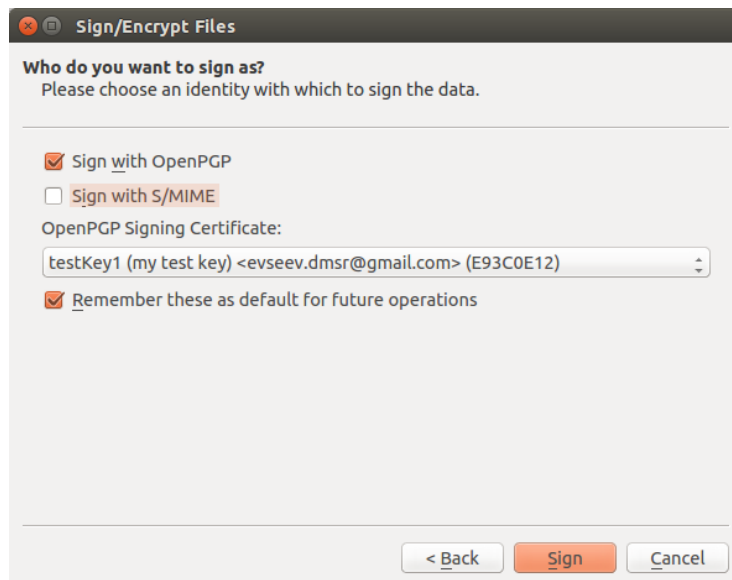


Рис. 8: Выбор стандарта и сертификата для ЦП.

Программа просит ввести пароль от сертификата. Вводим. Видим сообщение об успешном создании подписи на файл *readme.txt*, новый подписанный файл называется *readme.txt.sig* (рисунок 9).

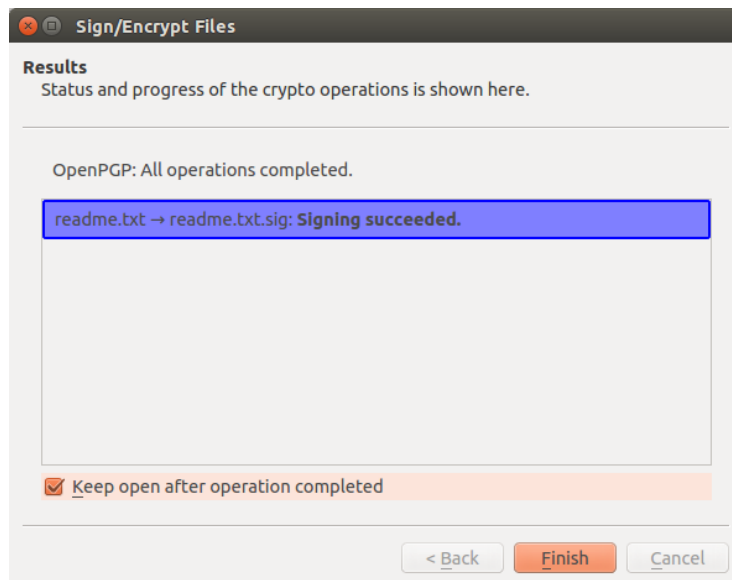


Рис. 9: Сообщение об успешном завершении.

3.4 Импорт сертификата и его подпись

Для импорта сертификата выполним команду *"FileImport Certificates"* и выберем необходимый файл типа *.asc* (рисунок 10).

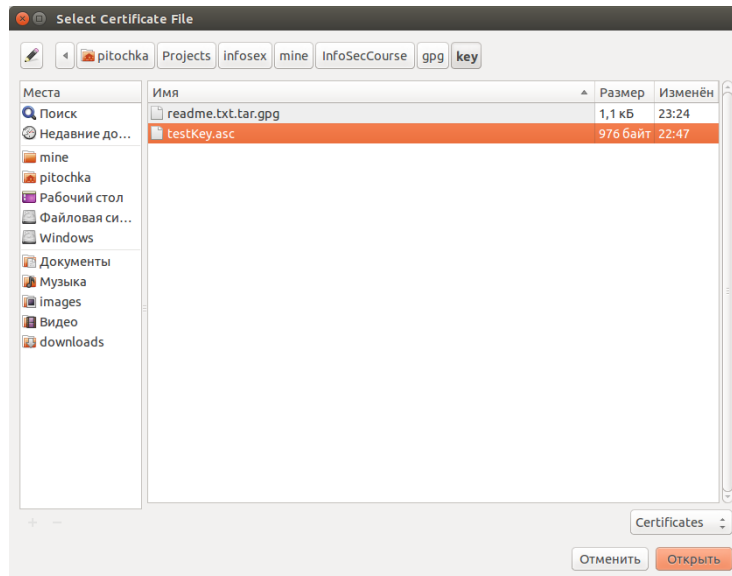


Рис. 10: Импорт сертификата.

Подпишем этот сертификат как в предыдущем пункте. Теперь мы храним файл *testKey.asc.sig*. Для проверки сертификата воспользуемся командой *"File> Decrypt/Verify Files"* и выберем подписанный ранее сертификат *testKey.asc.sig* (рисунок 11).

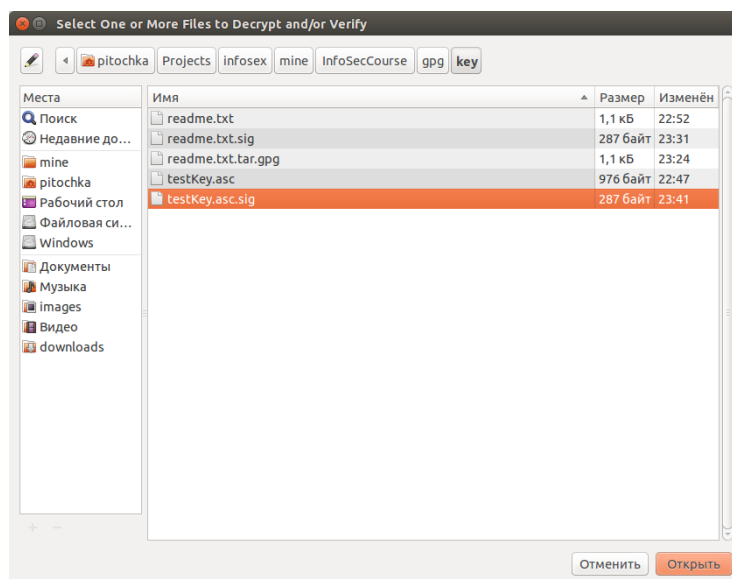


Рис. 11: Выбор сертификата для проверки.

Проверка показывает, кем была осуществлена подпись (рисунок 12).

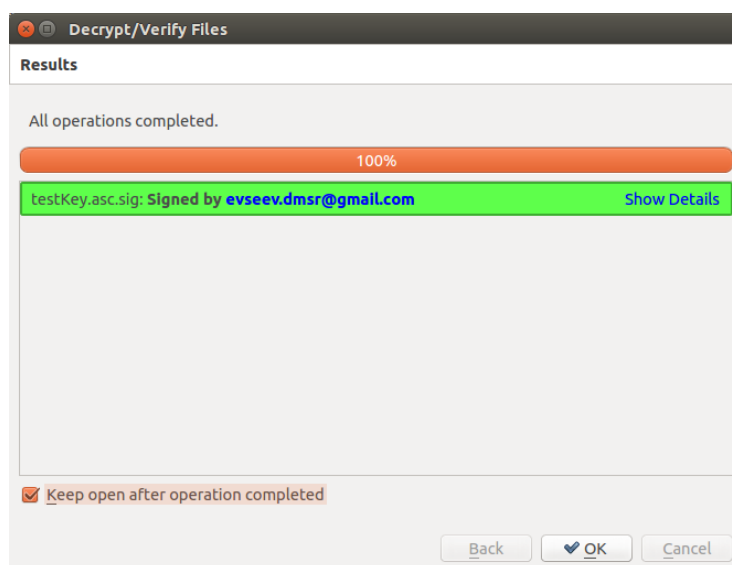


Рис. 12: Информация о подписи.

3.5 Расшифровка файла

Я с помощью моего ключа зашифровал документ и пытаюсь расшифровать *readme.txt.gpg*. Командой *"File > "Decrypt/Verify Files"* расшифруем документ. После расшифровки видим сообщение (рисунок 13), также появился

файл *readme.txt*, который можно прочитать.

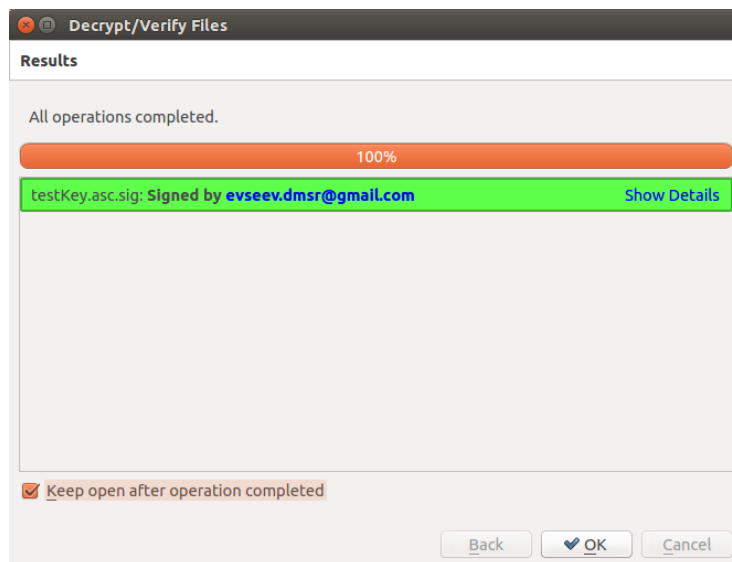


Рис. 13: Сообщение о расшифровке.

4 Использование GNU Privacy handbook

Операции, проделанные с помощью графического интерфейса Kleopatra, можно повторить используя лишь консоль.

Создание нового ключа производится с помощью команды (Рисунок 14, 15) `gpg --gen-key`

```
pitochka@pitochka-HP-Pavilion-15-Notebook-PC: ~/Projects/Infosex/victor/Infosex
Ключ не имеет ограничения срока действительности
Все верно? (y/N) y

Для идентификации Вашего ключа необходим User ID
Программа создаст его из Вашего имени, комментария и адреса e-mail в виде:
    "Baba Yaga (pensioner) <yaga@deepforest.ru>"

Ваше настоящее имя: Dmitriy
Email-адрес: evseev.dmsr@gmail.com
Комментарий:
Вы выбрали следующий User ID:
    "Dmitriy <evseev.dmsr@gmail.com>"

Сменить (N)Имя, (C)Комментарий, (E)email-адрес или (O)Принять/(Q)Выход?
o
Для защиты секретного ключа необходим пароль.

Необходимо сгенерировать много случайных чисел. Желательно, что бы Вы
выполняли некоторые другие действия (печать на клавиатуре, движения мыши
,
обращения к дискам) в процессе генерации; это даст генератору
случайных чисел возможность получить лучшую энтропию.

Недостаточно случайных чисел. Выполняйте какие-либо действия для того,
чтобы ОС могла получить больше случайных данных! (Необходимо ещё 166 байт)
.....+++++
.+++++
Необходимо сгенерировать много случайных чисел. Желательно, что бы Вы
выполняли некоторые другие действия (печать на клавиатуре, движения мыши
,
обращения к дискам) в процессе генерации; это даст генератору
случайных чисел возможность получить лучшую энтропию.

Недостаточно случайных чисел. Выполняйте какие-либо действия для того,
чтобы ОС могла получить больше случайных данных! (Необходимо ещё 55 байт)
)
```

Рис. 14: Создание ключа.

Для просмотра имеющихся в системе ключей необходимо воспользоваться командой (список ключей отображен на Рисунке 16) *gpg -list-keys*

```
pitochka@pitochka-HP-Pavilion-15-Notebook-PC: ~/Projects/infosex/victor/InfoSecCourse$
обращения к дискам) в процессе генерации; это даст генератору
случайных чисел возможность получить лучшую энтропию.

Недостаточно случайных чисел. Выполняйте какие-либо действия для того,
чтобы ОС могла получить больше случайных данных! (Необходимо ещё 166 байт)
.....+++++
+++++
Необходимо сгенерировать много случайных чисел. Желательно, что бы Вы
выполняли некоторые другие действия (печать на клавиатуре, движения мыши
,
обращения к дискам) в процессе генерации; это даст генератору
случайных чисел возможность получить лучшую энтропию.

Недостаточно случайных чисел. Выполняйте какие-либо действия для того,
чтобы ОС могла получить больше случайных данных! (Необходимо ещё 55 байт)
.....+++++
+++++
Недостаточно случайных чисел. Выполняйте какие-либо действия для того,
чтобы ОС могла получить больше случайных данных! (Необходимо ещё 21 байт)
+++++
gpg: ключ 10EA698B помечен как абсолютно доверяемый.
открытый и закрытый ключи созданы и подписаны.

gpg: проверка таблицы доверий
gpg: 3 ограниченных необходимо, 1 выполненных необходимо, PGP модель доверия
gpg: глубина: 0 корректных: 2 подписанных: 0 доверия: 0-, 0q, 0n, 0m, 0f, 2u
pub 2048R/10EA698B 2016-06-20
Отпечаток ключа = 7387 F38F 990A 7376 C170 3EBF EB4B C97C 10EA 69
8B
uid Dmitriy <evseev.dmsr@gmail.com>
sub 2048R/827BE95F 2016-06-20

pitochka@pitochka-HP-Pavilion-15-Notebook-PC:~/Projects/infosex/victor/InfoSecCourse$
```

Рис. 15: Создание ключа.

Экспорт ключей возможен с помощью команды *gpg -export* с различными опциями (например, вывод в файл, вид представления, выбор ключа для экспорта).

Для импорта - *gpg -import [filename]*

5 Вывод

В ходе лабораторной работы, используя пакет Gpg4win, я научилась создавать собственные ключевые пары и сертификаты на них; подписывать файлы и проверять подпись, а также зашифровывать и расшифровывать документы с помощью собственного сертификата или стороннего. Вышеперечисленные действия легко произвести как из графической оболочки Kleopatra.

```
pitochka@pitochka-HP-Pavilion-15-Notebook-PC: ~/Projects/infosec/victor/infosecCourse$ gpg --list-keys
/home/pitochka/.gnupg/pubring.gpg
-----
pub  2048R/E93C0E12 2016-05-15
uid          testKey1 (my test key) <evseev.dnsr@gmail.com>
pub  2048R/10EA698B 2016-06-20
uid          Dmitriy <evseev.dnsr@gmail.com>
sub  2048R/827BE95F 2016-06-20

pitochka@pitochka-HP-Pavilion-15-Notebook-PC: ~/Projects/infosec/victor/infosecCourse$
```

Рис. 16: Список ключей.