

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

Никитина Анна

28 февраля 2016 г.

Оглавление

1	Цель работы	2
2	Описание работы	2
3	Ход работы	3
3.1	Создание ключевой пары openPGP	3
3.2	Экспорт сертификата	4
3.3	Поставить ЭЦП на файл	5
3.4	Импорт сертификата и его подпись	7
3.5	Работа с чужим сертификатом	9
3.6	Использование GNU Privacy handbook	10

1 Цель работы

Научиться создавать сертификаты, шифровать файлы и ставить ЭЦП.

2 Описание работы

Шифрование — обратимое преобразование информации в целях сокрытия от неавторизованных лиц, с предоставлением, в это же время, авторизованным пользователям доступа к ней. Одним из способов шифрования является ЭЦП.

Электронная цифровая подпись (ЭЦП) — реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа (неотказуемость).

При выполнении лабораторной работы для шифрования и создания ЭЦП используется пакет Gpg4win. Он включает в себя:

- версию GnuPG — свободная программа для шифрования информации и создания электронных цифровых подписей;
- Kleopatra (менеджер сертификатов для OpenPGP и X.509);
- GPG (альтернативный менеджер сертификатов (GNU) для OpenPGP и X.509);
- другие компоненты.

3 Ход работы

Дальнейшие действия будут выполнены в графической оболочке "Kleopatra".

3.1 Создание ключевой пары OpenPGP

Для создания новой ключевой пары OpenPGP выполним команду *"File -> New Certificate"*. После чего необходимо ввести персональную информацию: имя сертификата, адрес электронной почты пользователя (рисунок 1).

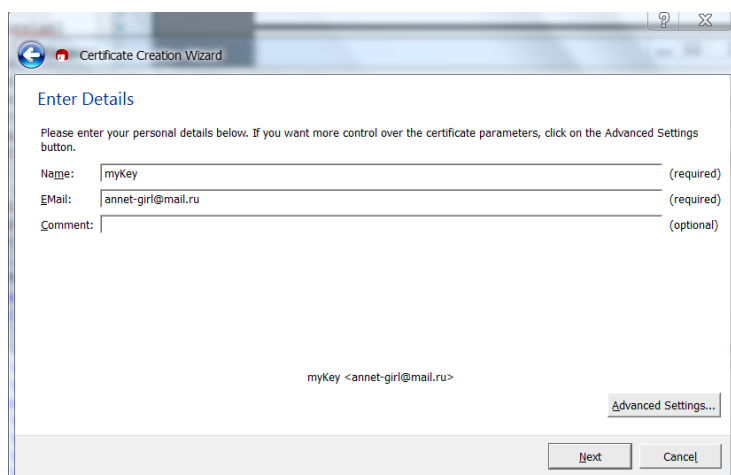
The screenshot shows a window titled 'Certificate Creation Wizard'. The main heading is 'Enter Details'. Below it, a message says: 'Please enter your personal details below. If you want more control over the certificate parameters, click on the Advanced Settings button.' There are three input fields: 'Name:' with the value 'myKey' (marked as required), 'Email:' with the value 'annet-girl@mail.ru' (marked as required), and 'Comment:' (marked as optional). Below these fields, the text 'myKey <annet-girl@mail.ru>' is displayed. At the bottom right, there is an 'Advanced Settings...' button. At the very bottom, there are 'Next' and 'Cancel' buttons.

Рис. 1: Окно для ввода персональных данных.

Далее подтвердим персональные данные, нажав кнопку *"Create Key"* (рисунок 2).

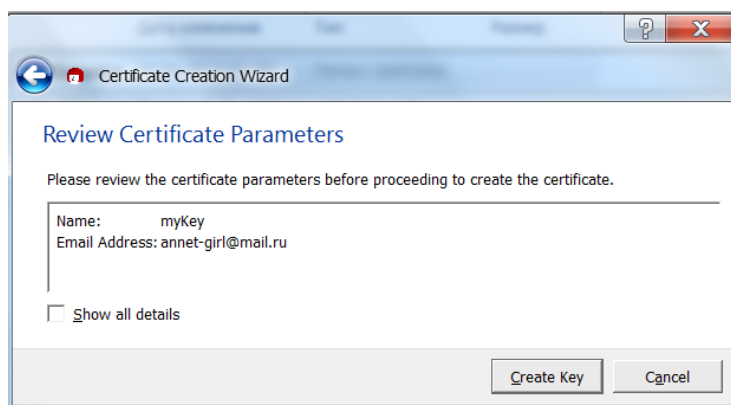
The screenshot shows the same 'Certificate Creation Wizard' window, but at the 'Review Certificate Parameters' step. The heading is 'Review Certificate Parameters'. A message says: 'Please review the certificate parameters before proceeding to create the certificate.' Below this, the 'Name:' is 'myKey' and 'Email Address:' is 'annet-girl@mail.ru'. There is a checkbox labeled 'Show all details' which is currently unchecked. At the bottom right, there are 'Create Key' and 'Cancel' buttons.

Рис. 2: Окно создания ключа.

После необходимо дважды ввести фразу-пароль (рисунок 3).
Сертификат успешно создан (рисунок 4).

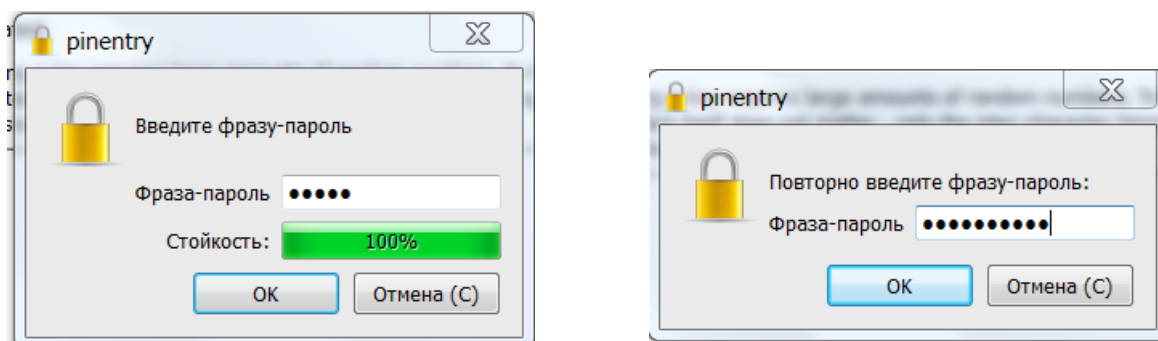


Рис. 3: Окно ввода и подтверждения фразы-пароля

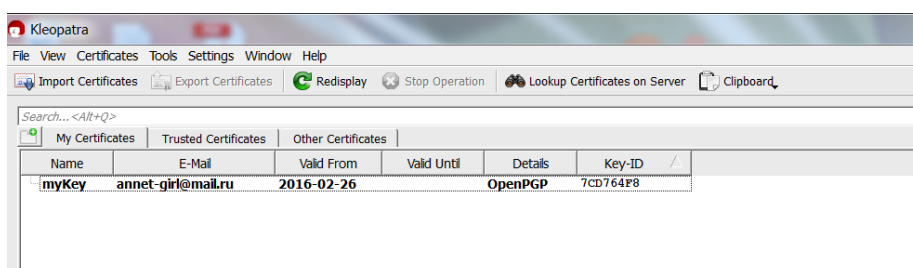


Рис. 4: Созданный сертификат

3.2 Экспорт сертификата

Для экспорта сертификата выполним команду *"File -> Export Certificate"*. После чего введем имя файла *key.asc* (рисунок 5).

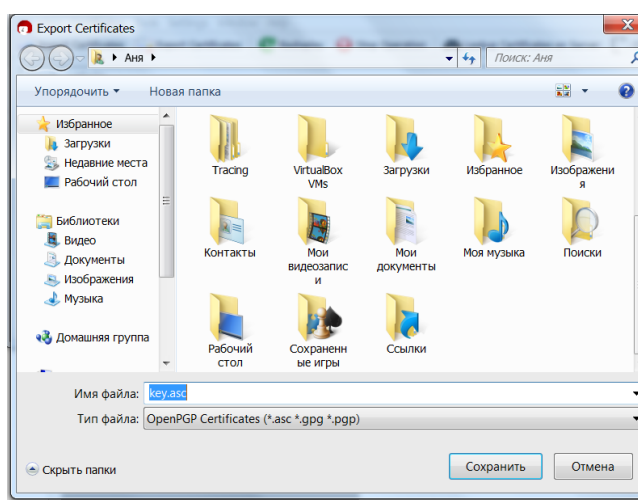


Рис. 5: Экспорт сертификата

3.3 Поставить ЭЦП на файл

Для того, что бы поставить ЭЦП на файл выполним команду "*File -> Sign/Encrypt Files*" и выберем файл, на который необходимо поставить ЭЦП. В нашем случае это *1.PNG* (рисунок 6).

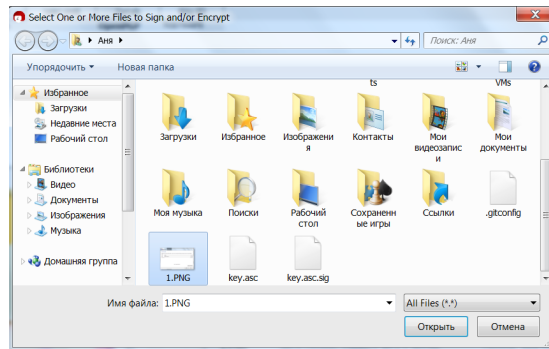


Рис. 6: Выбор файла

После выберем одно из трех предложенных действий.

- Sign and Encrypt
- Encrypt
- Sign

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

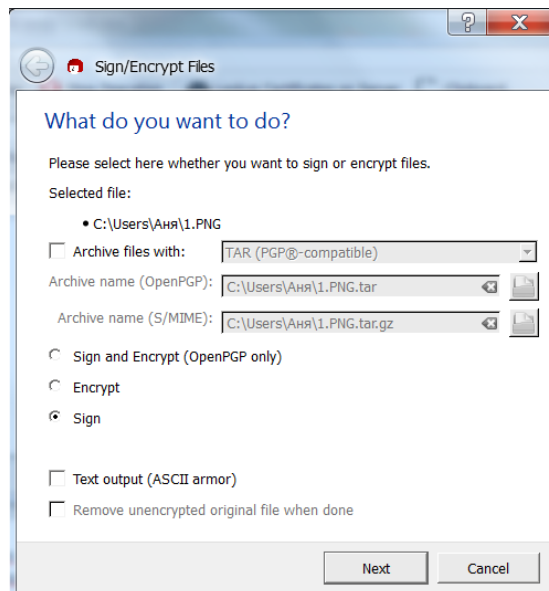


Рис. 7: Поставить ЭЦП на файл

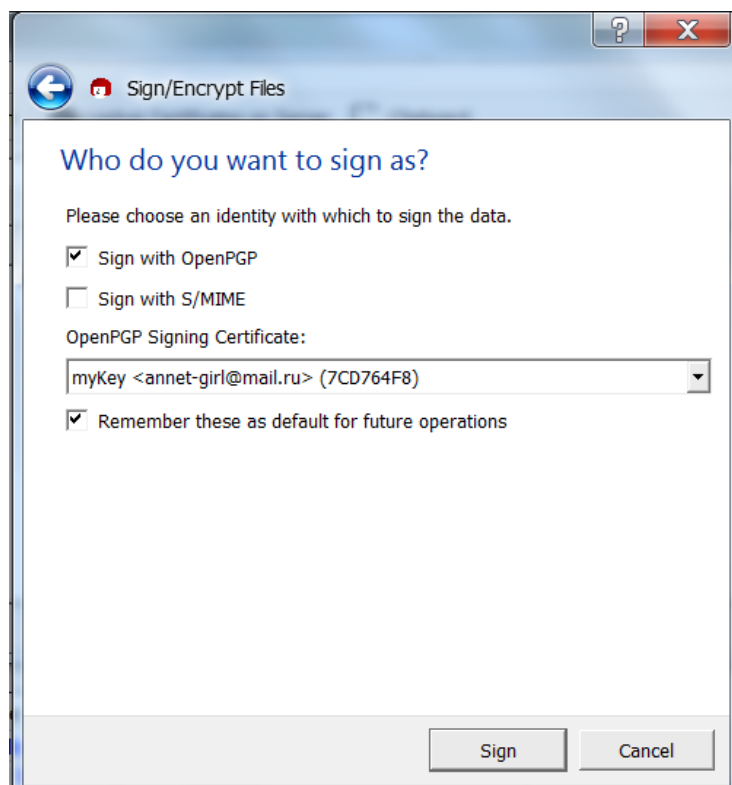


Рис. 8: Выбор стандарта и сертификата для ЭЦП

Выберем для подписи стандарт OpenPGP и сертификат, созданный ранее (рисунок 8).

Введем пароль (рисунок 9).

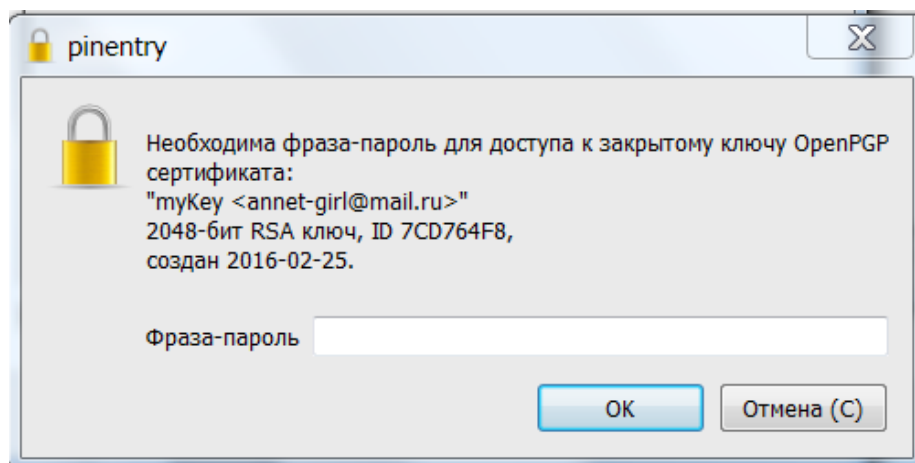


Рис. 9: Ввод пароля

Видим сообщение об успешном создании подписи на файл *1.PNG*, новый подписанный файл называется *1.PNG.sig* (рисунок 10).

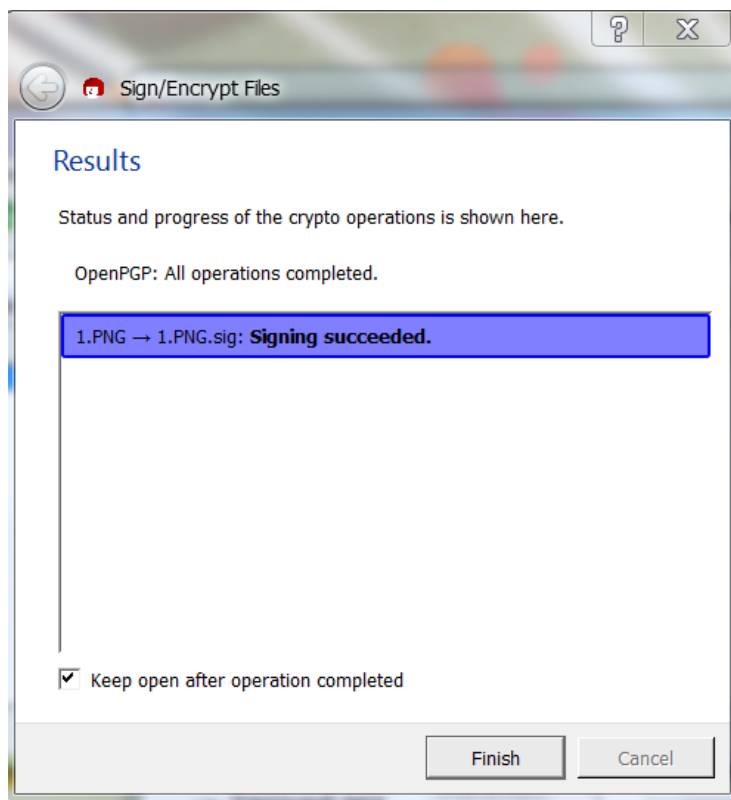


Рис. 10: Успешная подпись файла

3.4 Импорт сертификата и его подпись

Для импорта сертификата выполним команду *File -> Import Certificates* и выберем необходимый файл типа *.asc* (рисунок 11).

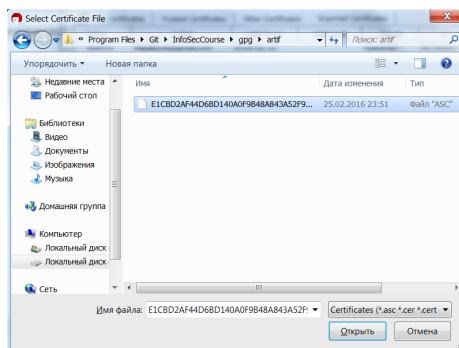


Рис. 11: Выбор файла для импорта

После чего импортированный сертификат появится в рабочем пространстве (рисунок 12).

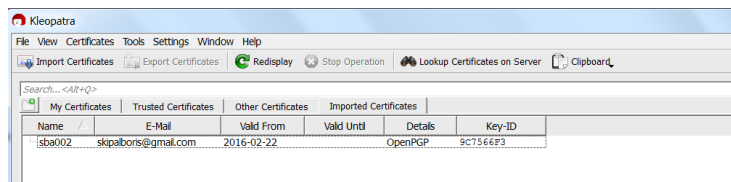


Рис. 12: Успешный импорт сертификата

Подпишем его, как это было описано выше (рисунок 13).

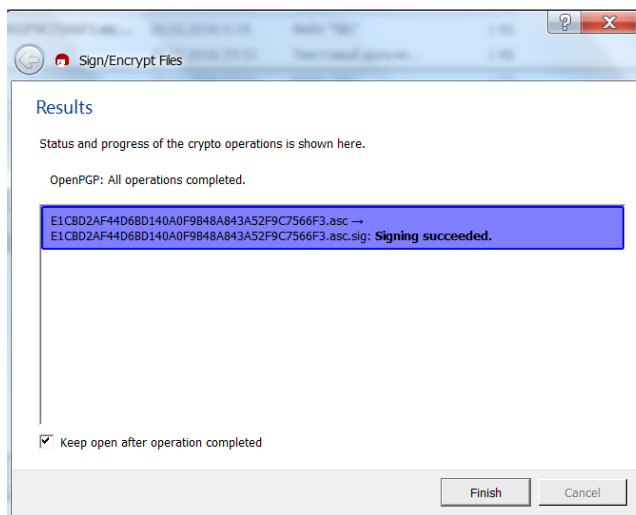


Рис. 13: Подпись импортированного сертификата

Для проверки воспользуемся командой *File -> Decrypt/Verify Files* и выберем подписанный ранее сертификат типа *.asc/sig* (рисунок 14).

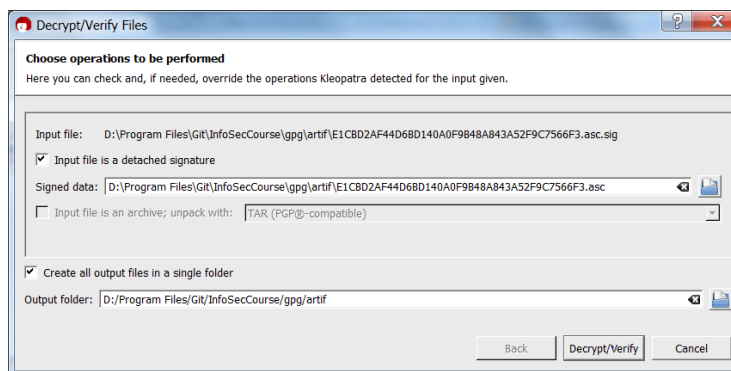


Рис. 14: Проверка подписи

Проверка показывает, кем была осуществлена подпись (рисунок 15).

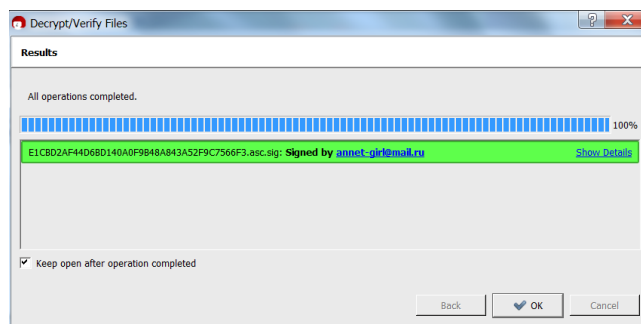


Рис. 15: Проверка подписи

3.5 Работа с чужим сертификатом

С помощью чужого сертификата зашифруем и подпишем какой-либо документ, например файл *Боря.txt*. В свою очередь коллега, которому принадлежит этот сертификат, расшифрует переданный документ *Боря.txt.gpg*, используя свой пароль.

Прделаем действия в другую сторону. Коллега с помощью моего ключа зашифровал документ и передал его мне *Ann.txt.gpg*. Командой *File -> Decrypt/Verify Files* расшифруем документ. После ввода верного пароля видим окно с сообщением об удачном расшифровании файла. (рисунок 16), также появился файл *Ann.txt*, который можно прочитать.

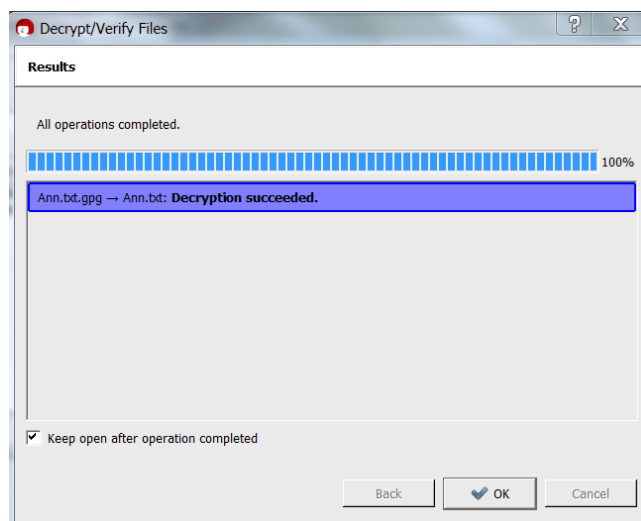


Рис. 16: Успешное расшифрование файла

3.6 Использование GNU Privacy handbook

С помощью GNU Privacy handbook сделаем некоторые действия по использованию `gpg` через командную строку.

Для создания ключевой пары введем в консоли команду `gpg --gen-key`. Далее выберем тип ключа, его размер, срок действия, укажем ID пользователя, электронную почту, введем пароль, после чего создастся ключевая пара. Полный лог вышеперечисленных действий приведен в файле `log.txt`.

Создали ключ типа RSA и DSA, размером 1024, срок действия которого не ограничен.

Для создания сертификата воспользуемся командой `gpg --output key_console.asc --gen-revoke anna_nik`, в которой укажем выходной файл сертификата и ключевую пару. После ответа на необходимые вопросы сертификат будет создан (рисунок 17).

```
C:\Users\Аня>gpg --output key_console.asc --gen-revoke anna_nik
sec 1024R/2C88B9FE 2016-02-28 anna_nik <annet-girl@mail.ru>

Создать сертификат отзыва данного ключа? (y/N) y
Укажите причину отзыва:
  0 = Причина не указана
  1 = Ключ был раскрыт
  2 = Ключ заменен другим
  3 = Ключ больше не используется
  0 = Отмена
(Скорее всего, Вы здесь выберете 1)
Ваше решение? 0
Введите необязательное пояснение; закончите пустой строкой:
>
Причина отзыва: Причина не указана
(Пояснения отсутствуют)
Все правильно? (y/N) y

Необходима фраза-пароль для доступа к закрытому ключу пользователя: "anna_nik <annet-girl@mail.ru>"
1024-битный ключ RSA, ID 2C88B9FE, создан 2016-02-28

gpg: Неверная фраза-пароль; попробуйте еще раз ...

Необходима фраза-пароль для доступа к закрытому ключу пользователя: "anna_nik <annet-girl@mail.ru>"
1024-битный ключ RSA, ID 2C88B9FE, создан 2016-02-28

Для вывода использован текстовый формат ASCII.
Сертификат отзыва создан.

Поместите его в надежное место; если посторонний получит доступ
к данному сертификату, он может использовать его, чтобы сделать
Ваш ключ непригодным к использованию. Можно распечатать данный
сертификат и спрятать подальше на случай, если Ваш основной
носитель будет поврежден, но будьте осторожны: система печати
Вашей машины может сохранить данные и сделать их доступными для других!
```

Рис. 17: Создание сертификата

Для просмотра списка созданных ключей введем команду `gpg --list-key`. Видим в списке все ключи, созданные или импортированные как в графической оболочке, так и в консоли (рисунок 18).

Для того, чтобы отправить свой открытый ключ корреспонденту необходимо его экспортировать командой `gpg --output anna_nik.gpg --export anna_nik`.

Для подписи документа используем команду `gpg --output 2.PNG.sin --sign 2.PNG`. После ввода пароля создается новый подписанный документ `2.PNG.sin` (рисунок 19).

```

C:\Users\Аня>gpg --list-key
C:/Users/Аня/AppData/Roaming/gnupg/pubring.gpg
-----
pub  2048R/7CD764F8  2016-02-25
uid  [абсолютное]  myKey <annet-girl@mail.ru>
sub  2048R/4BECA6BD  2016-02-25

pub  2048R/9C7566F3  2016-02-22
uid  [неизвестно]  sba002 <skipalboris@gmail.com>
sub  2048R/00808598  2016-02-22

pub  1024R/2C88B9FE  2016-02-28
uid  [абсолютное]  anna_nik <annet-girl@mail.ru>
sub  1024R/5CC6B2DD  2016-02-28

```

Рис. 18: Список созданных ключей

```

C:\Users\Аня>gpg --output 2.PNG.sin --sign 2.PNG
Необходима фраза-пароль для доступа к закрытому ключу пользователя: "myKey <anne
t-girl@mail.ru>"
2048-битный ключ RSA, ID 7CD764F8, создан 2016-02-25

```

Рис. 19: Подпись документа

4 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**, так и из командной строки.