Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Методы и средства защиты информации

Отчет по лабораторной работе №1
Программа для шифрования и подписи GPG, пакет Gpg4win

Работу выполнил:
Косолапов С.А.
Группа: 53501/3
Преподаватель:
Вылегжанина К.Д.

Санкт-Петербург
2016

Содержание

1 Цель работы

Научиться создавать сертификаты, шифровать файлы и ставить ЭЦП.

2 Программа работы

- Изучить документацию, запустить графическую оболочку Kleopatra
- Создать ключевую пару OpenPGP (File → New Certificate)
- Экспортировать сетрификат (File → Export Certificate)
- Поставить ЭЦП на файл (File → Sign/Encrypt Files)
- Взять сертификат кого-либо из коллег, зашифровать и подписать для него какой-либо текст, предоставить свой сертификат, убедиться, что ему удалось получить открытый текст, проверить подпись
- Предыдущий пункт наоборот
- Используя GNU Privacy handbook (ссылка в материалах) потренироваться в использовании gpg через интерфейс командной строки, без использования графических оболочек.

3 Теоретическая информация

4 Ход выполнения работы

4.1 GPG и Оболочка Kleopatra

PGP - сокращение от Pretty Good Privacy, в переводе - ”достаточно хорошая секретность”, название программы , написанной Филиппом Циммерманном в 1991 году. Эта программа позволяла удобным образом использовать шифрование данных. За её написание Циммермана пытались судить в США, программу перекупали друг у друга разные фирмы, в данный момент PGP принадлежит фирме Symantec, и PGP является коммерческой программой.

Kleopatra - менеджер сертификатов для OpenPGP (одной из реализаций PGP) и x.509 с GUI. Есть реализации под многие ОС семейств Windows, Linux.

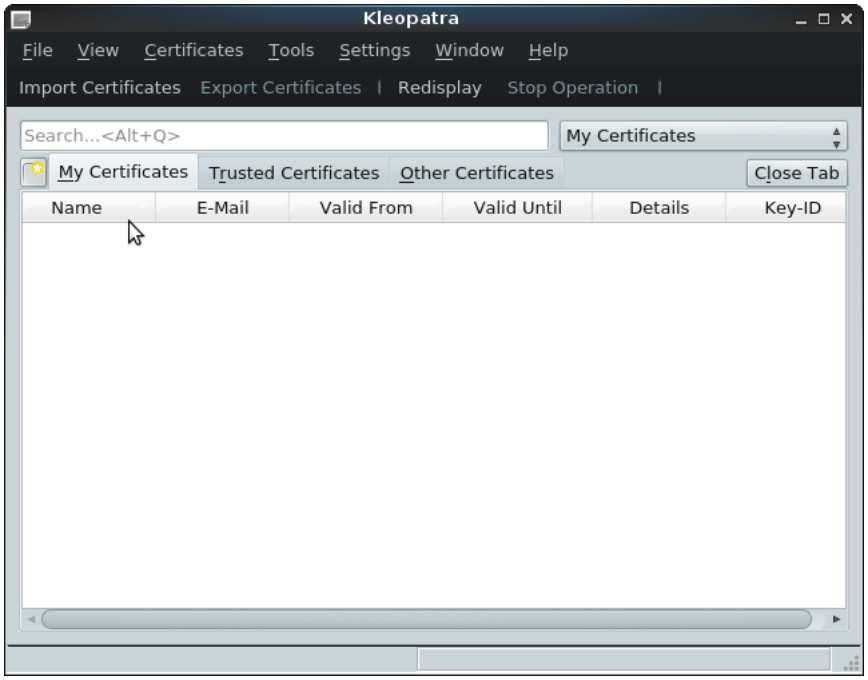

Рис. 1: Внешний вид программы Kleopatra (без сертификатов)

Как видим, основную часть окна занимает ListBox с отображаемыми сертификатами. Можем выбрать между "Моими сертификатами "Доверенными сертификатами"и "Другими сертификатами". Ниже меню расположены кнопки, позволяющие произвести импорт/экспорт сертификата, обновить ("Повторно отобразить") сертификаты и "Остановить операцию".

4.2 Создание нового сертификата

Создать новый сертификат можно, нажав клавиши Ctrl+N или через меню File -> New Sertificate. В этом случае появится окно, позволяющее выбрать, что же, собственно, мы хотим создать - GPG или x509 сертификат.

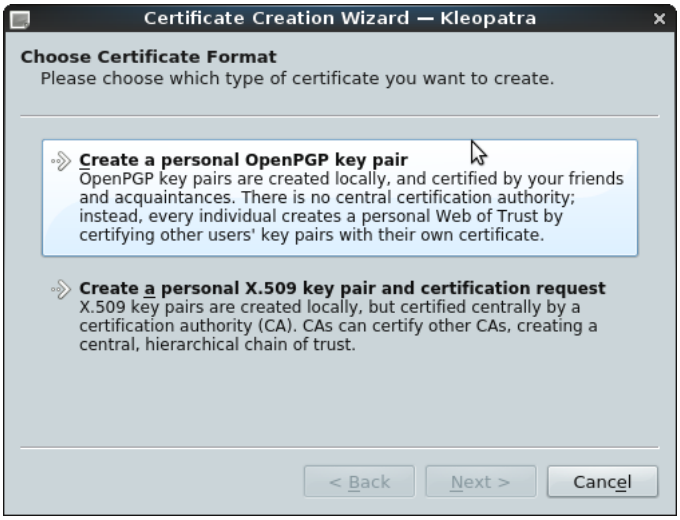

Рис. 2: Окно выбора типа сертификата при создании

Выбираем создание пары ключей OpenPGP (первый пункт).

Появляется окно, в котором нужно указать характеристики сертификата - владельца, его адрес электронной почты и, при необходимости, комментарий.

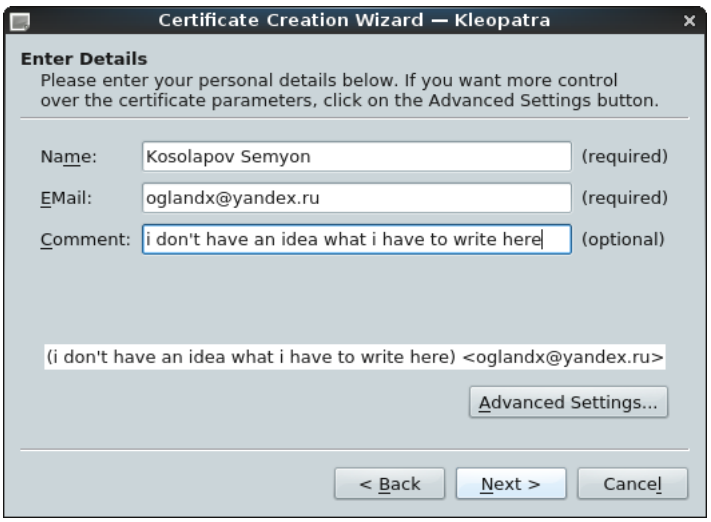

Рис. 3: Окно ввода персональных данных при создании сертификата GPG

Также можно выбрать дополнительные настройки. Можно указать тип алгоритма шифрования (RSA/DSA/ECDSA) и его длину. В данном случае выбран стандартный вариант - 2048 бит RSA. Ниже можно указать цель использования сертификата и дату окончания его срока действия.

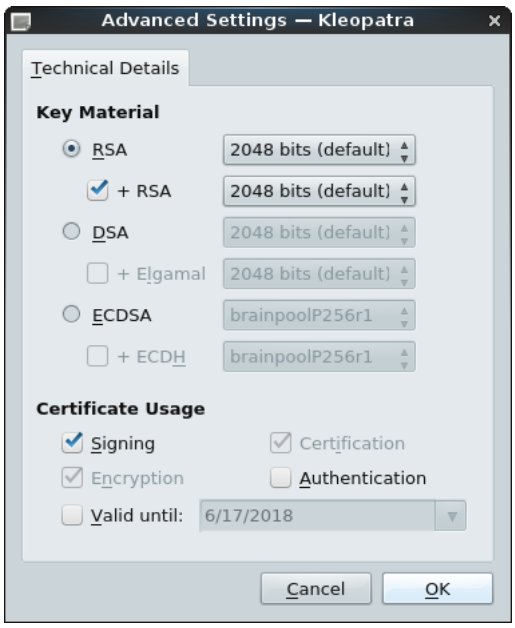

Рис. 4: Окно указания технических деталей GPG

Далее визард предоставляет возможность ещё раз просмотреть введённые параметры.

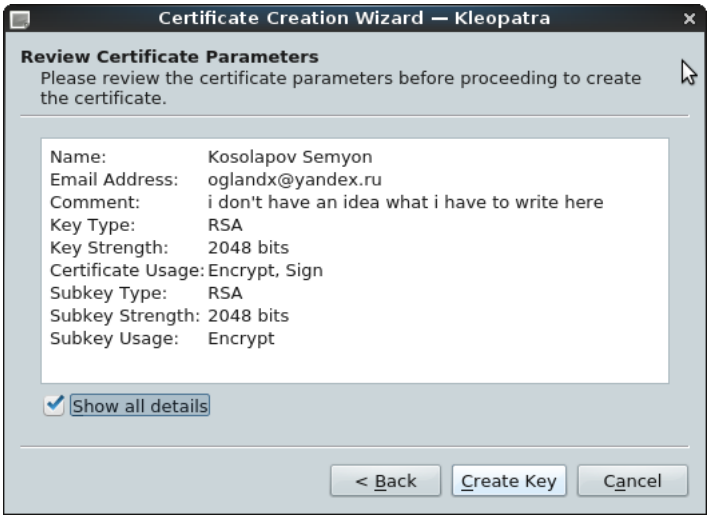

Рис. 5: Ревью параметров сертификата (с просмотром дополнительных параметров)

Далее производится создание пары ключей для сертификата.

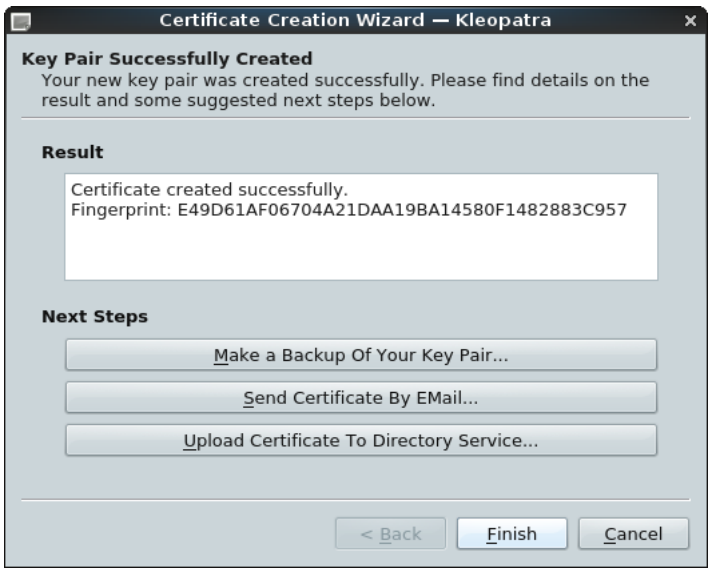

Рис. 6: Окно, демонстрирующее успешное создание ключа

Можем просмотреть ключи сертификата.

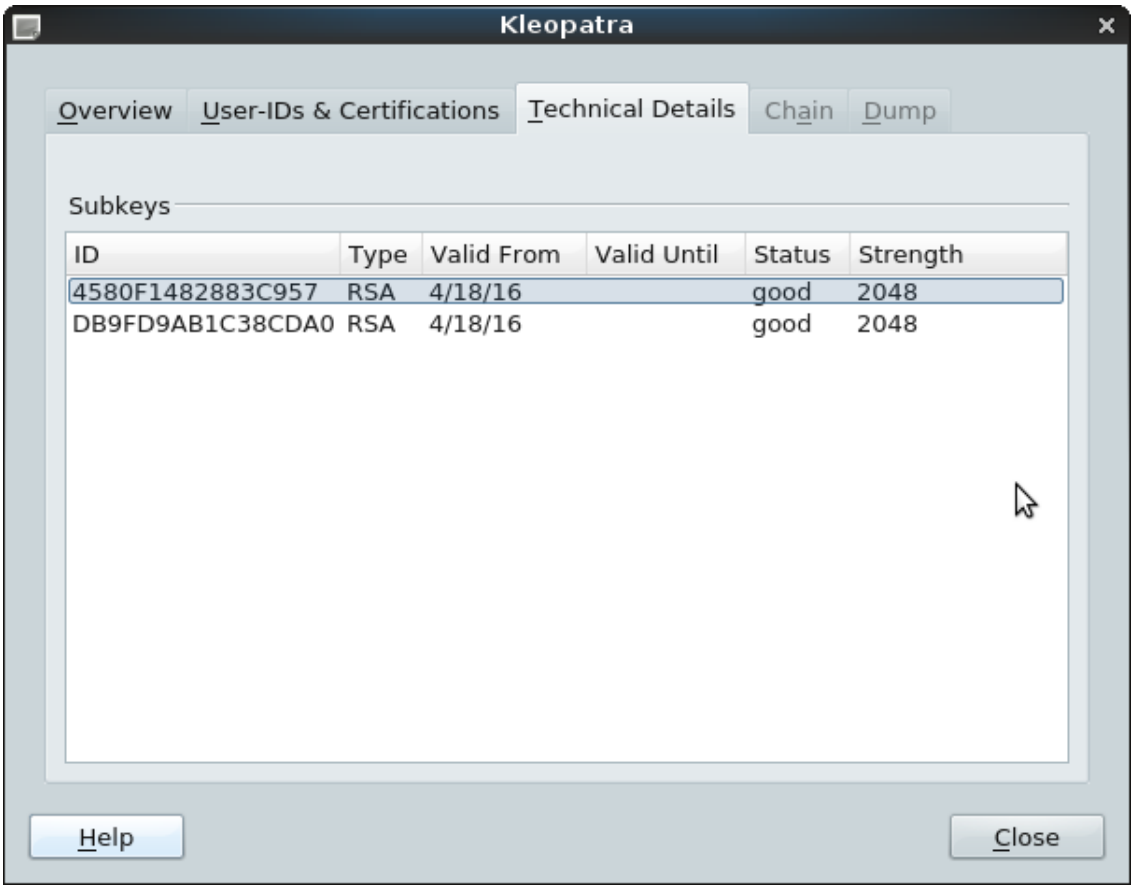

Рис. 7: Просмотр ключей сертификата

После создания сертификат стал отображаться в главном окне в списке сертификатов. Там можно просмотреть его характеристики, удалить и т.д.

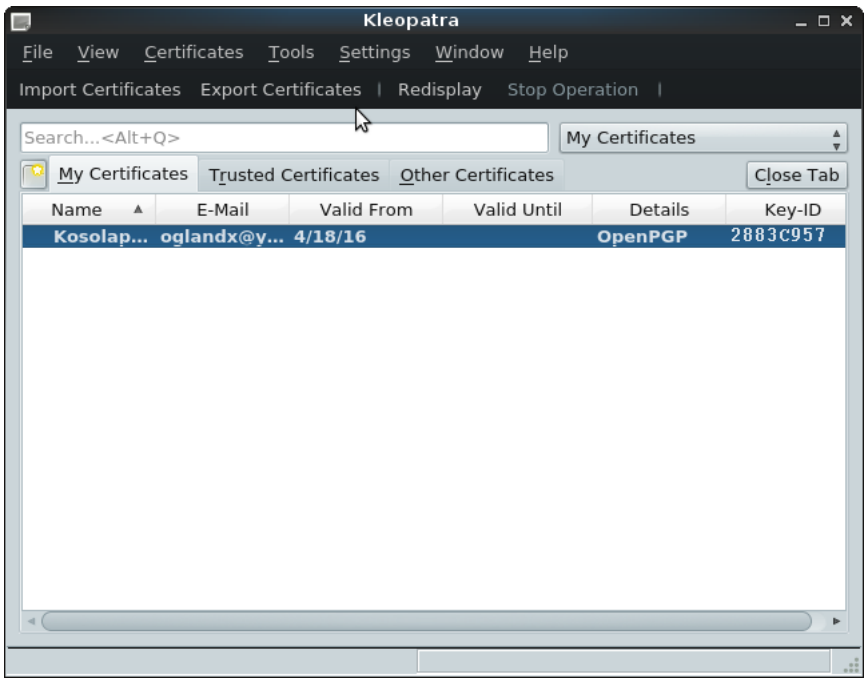

Рис. 8: Вид главного окна после создания сертификата

4.3 Экспорт сертификата

В после того, как сертификат был создан, можем его экспортировать. Для этого нужно нажать на кнопку "Экспортировать"в верхней части экрана. Появится диалог сохранения.

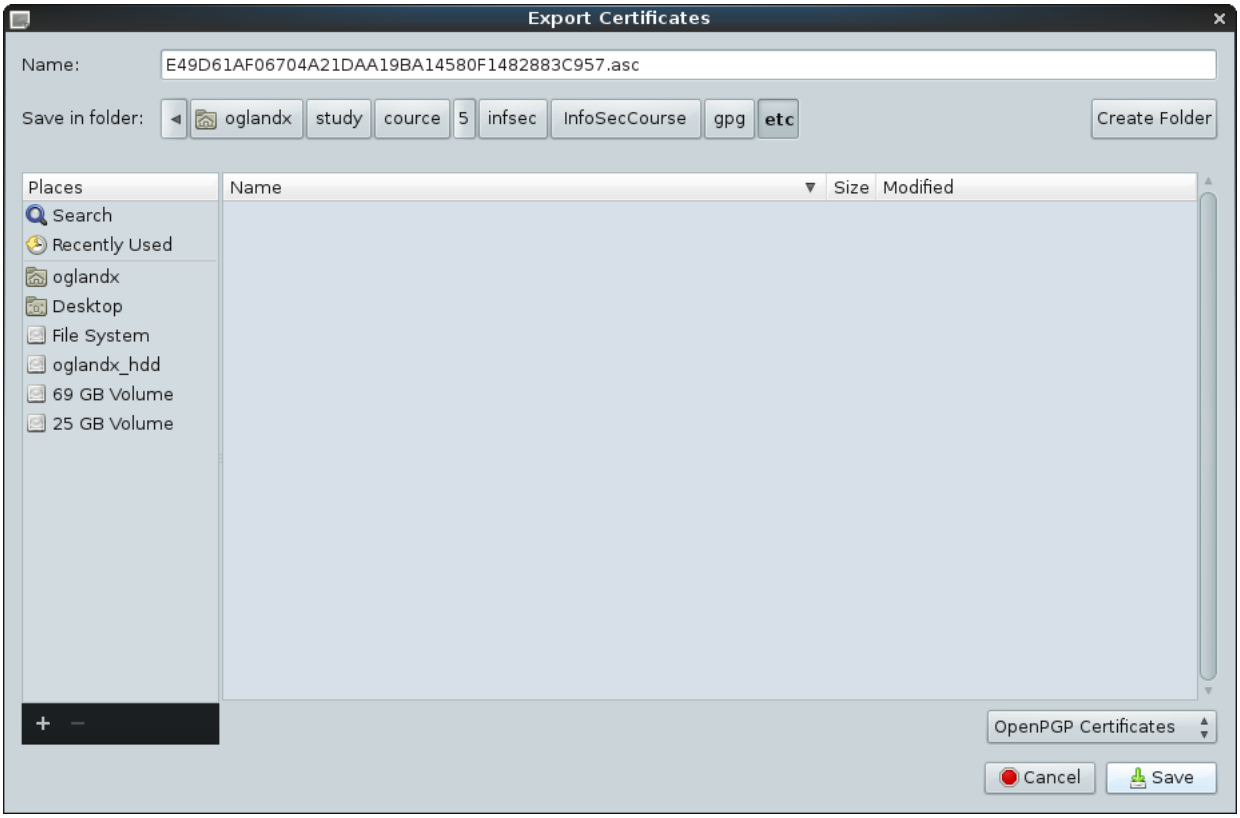

Рис. 9: Выбор директории для сохранения сертификата

4.4 Установка ЭЦП на файл

Далее можем установить ЭЦП на файл, который, для начала, нужно выбрать.

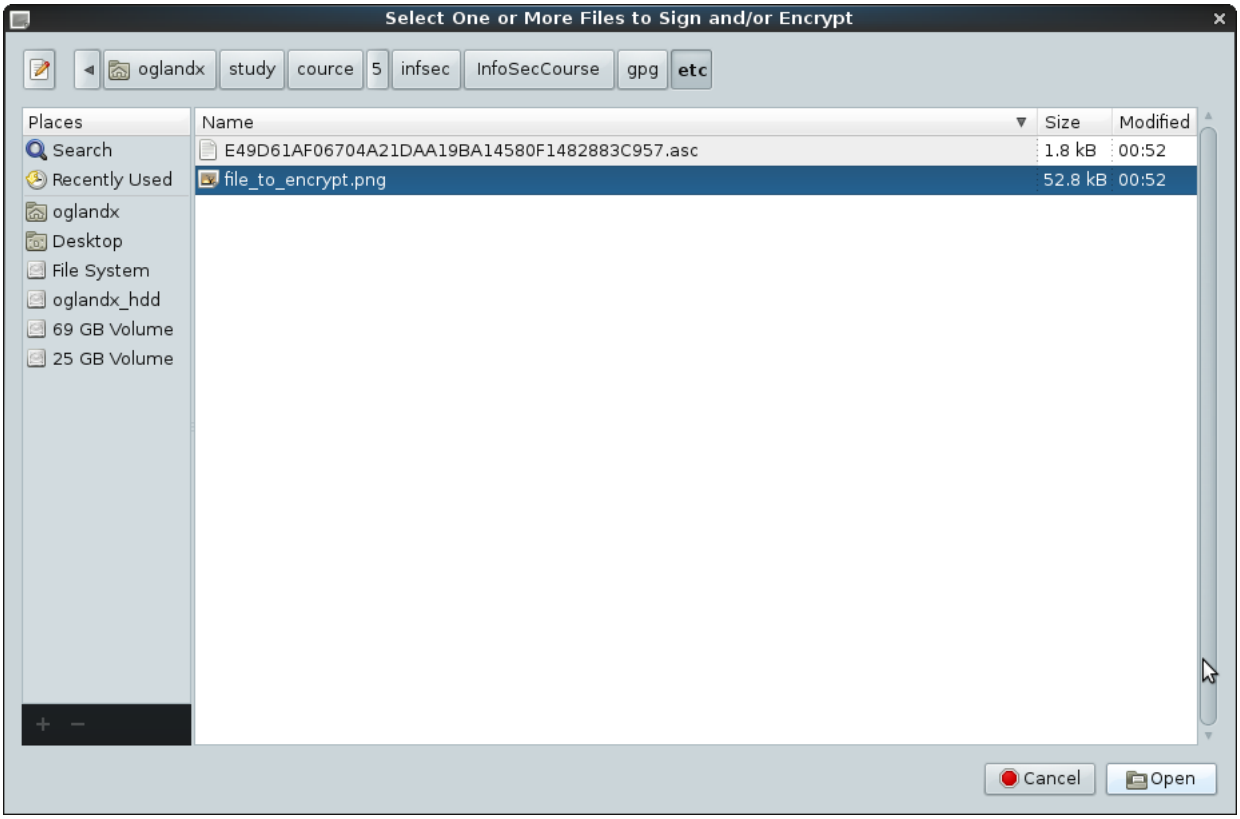

Рис. 10: Выбор файла для установки ЭЦП

После чего появится диалог, в котором можно выбрать действия, которые мы хотим произвести.

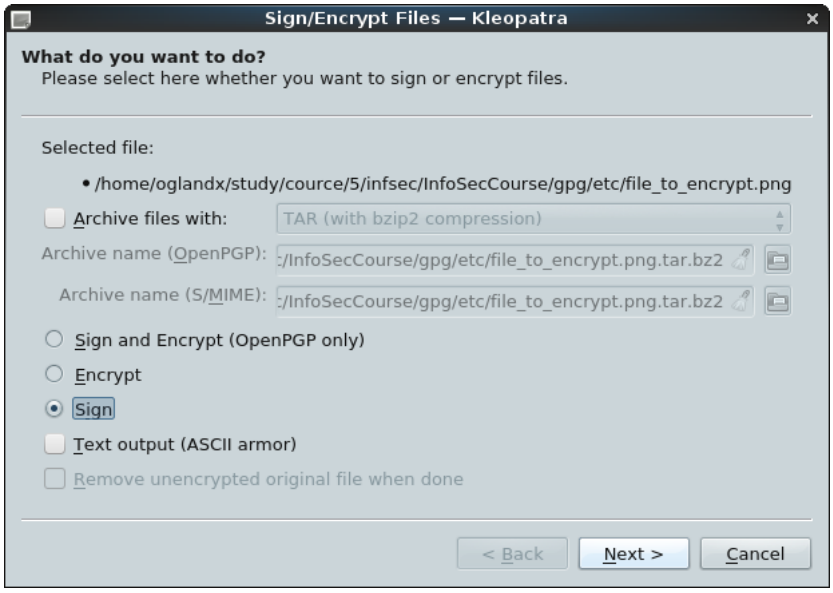

Рис. 11: Выбор параметров

Установим, что хотим подписать, после чего нажмём next.

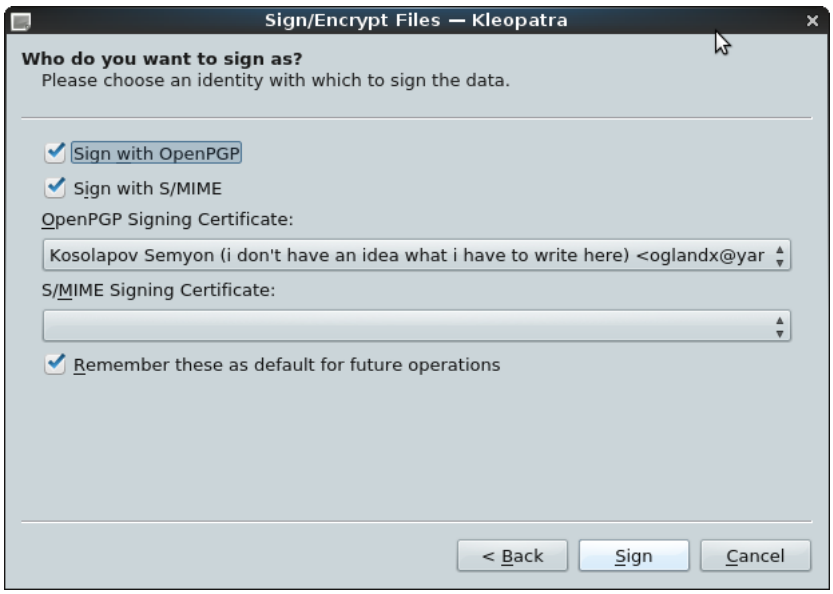

Рис. 12: Выбор параметров подписи

После выбора параметров подписи и ввода passphrase, получаем уведомление об успешности операции:

Рис. 13: Выбор параметров подписи

Можем посмотреть, что за файл был создан:

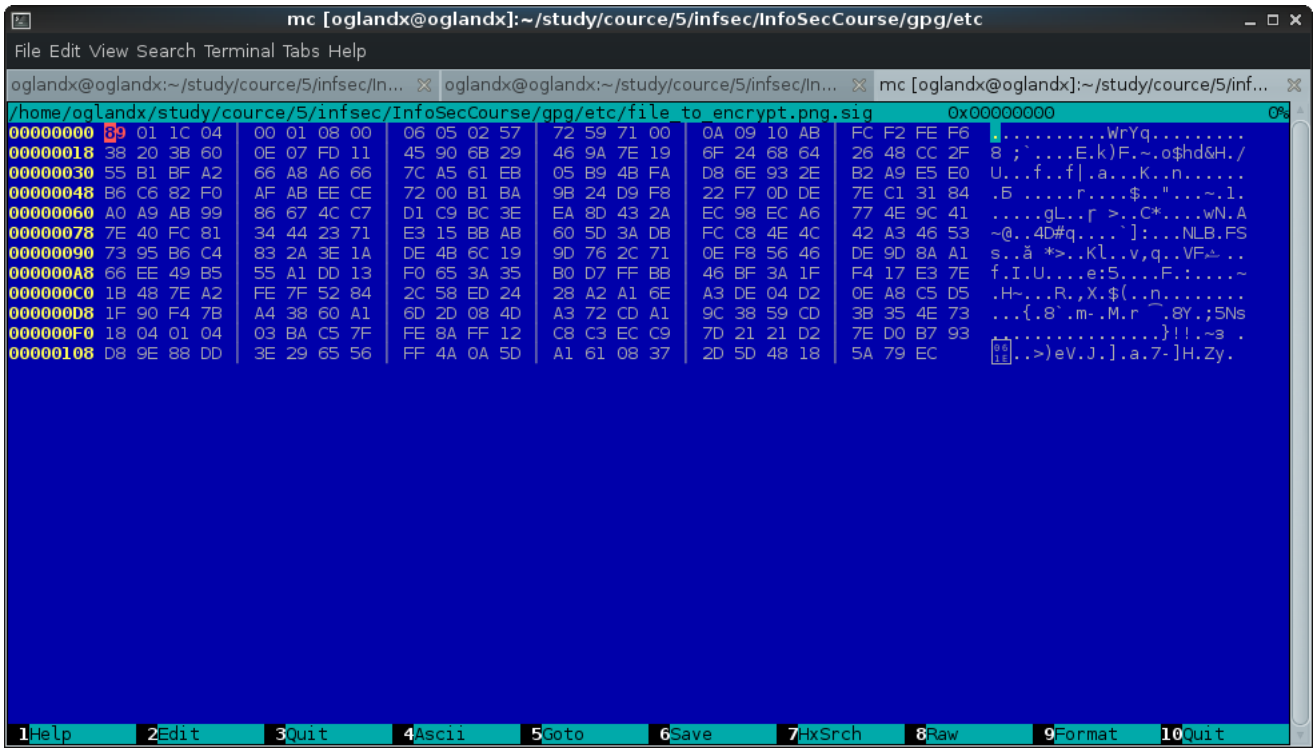

Рис. 14: Содержимое sig-файла

Как видим, файл имеет довольно небольшой размер. Оригинальный файл должен находиться в одной директории с файлом *.sig.

4.5 Шифрование файла

Так же, как и в предыдущем пункте, выбираем файл. После этого в диалоге выбираем encrypt вместо sign.

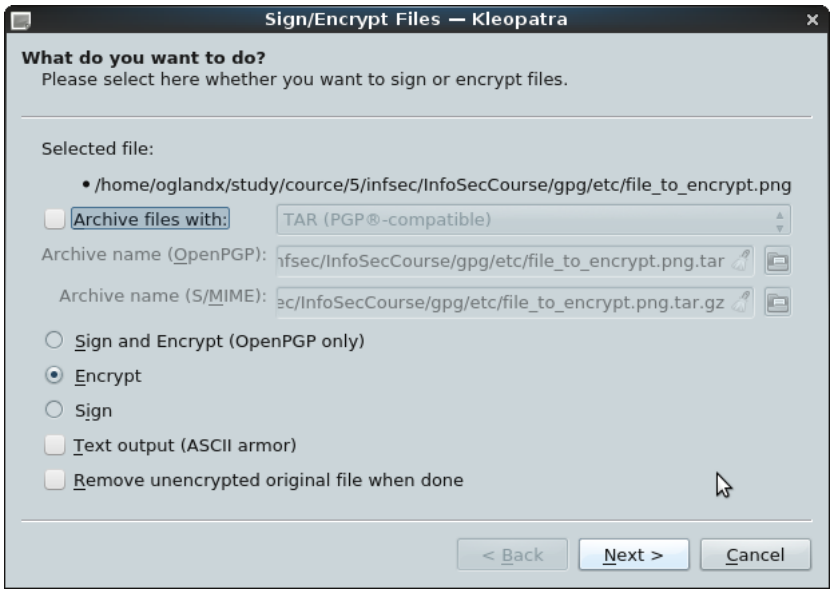

Рис. 15: Выбор опции шифрования

Далее предоставляется возможность выбрать сертификат.

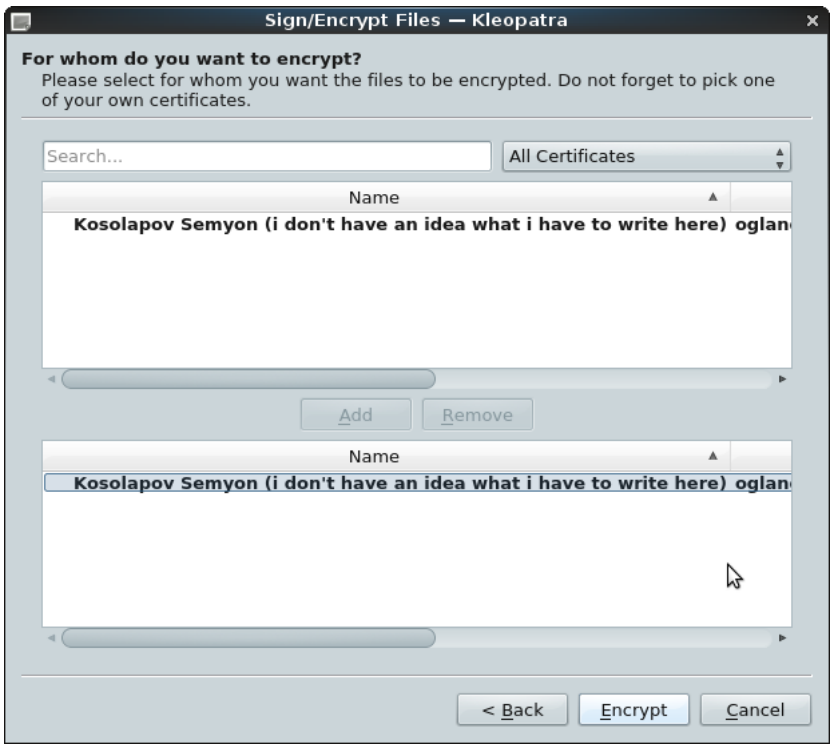

Рис. 16: Выбор сертификата

После чего в диалоге появляется сообщение об успешности проведённой операции.

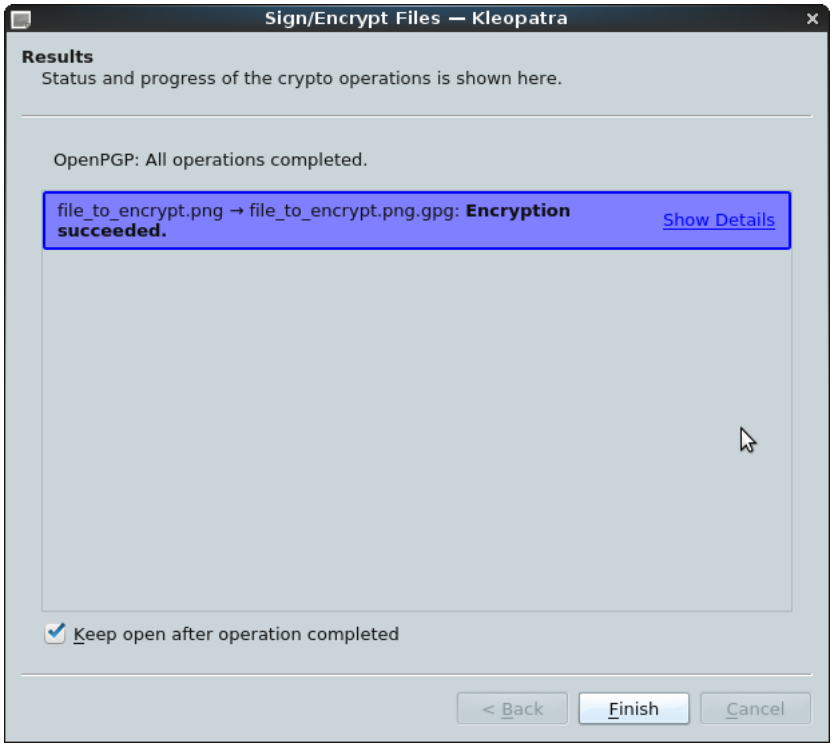

Рис. 17: Шифрование прошло успешно

В результате, получили файл с расширением *.gpg.

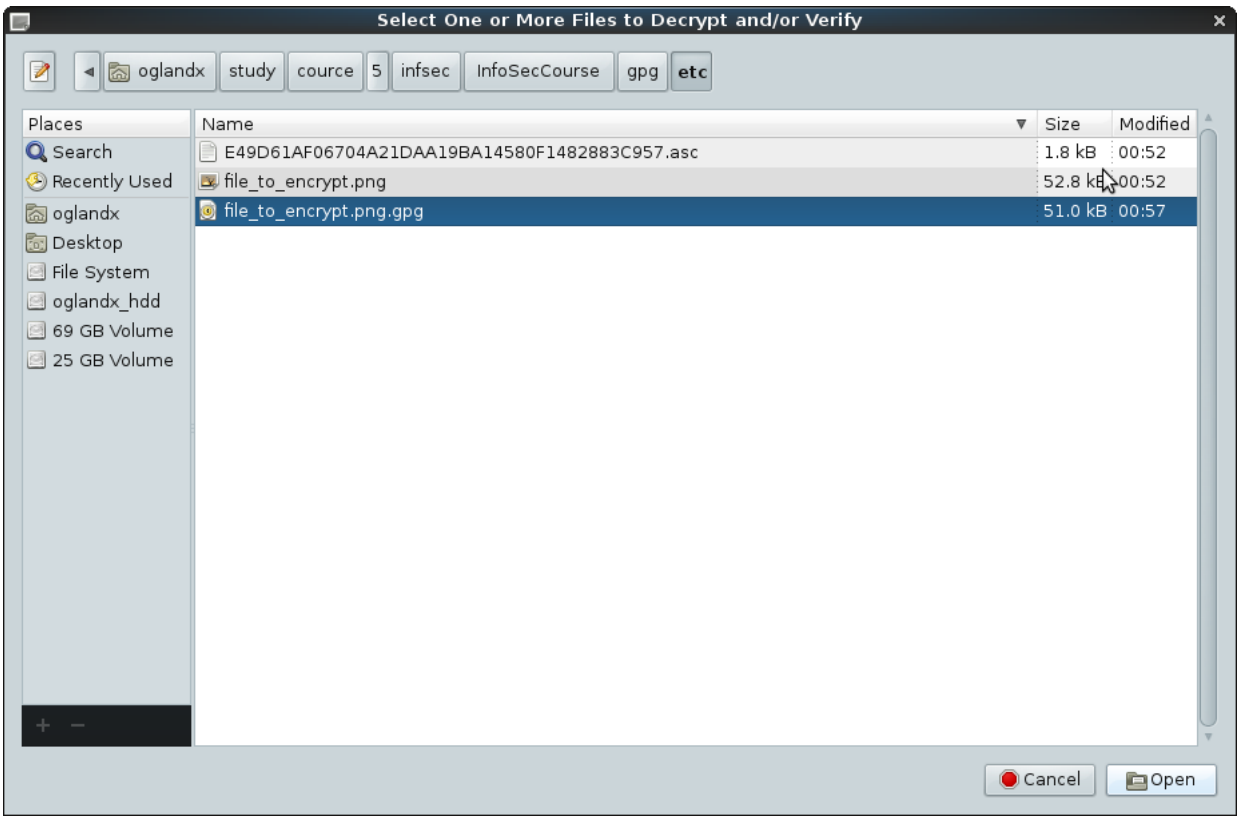

Рис. 18: Файл с расширением .gpg создан после шифрования

Далее можем попробовать расшифровать файл. Для этого выбираем Decrypt в диалоге Decrypt/Verify Files. Далее:

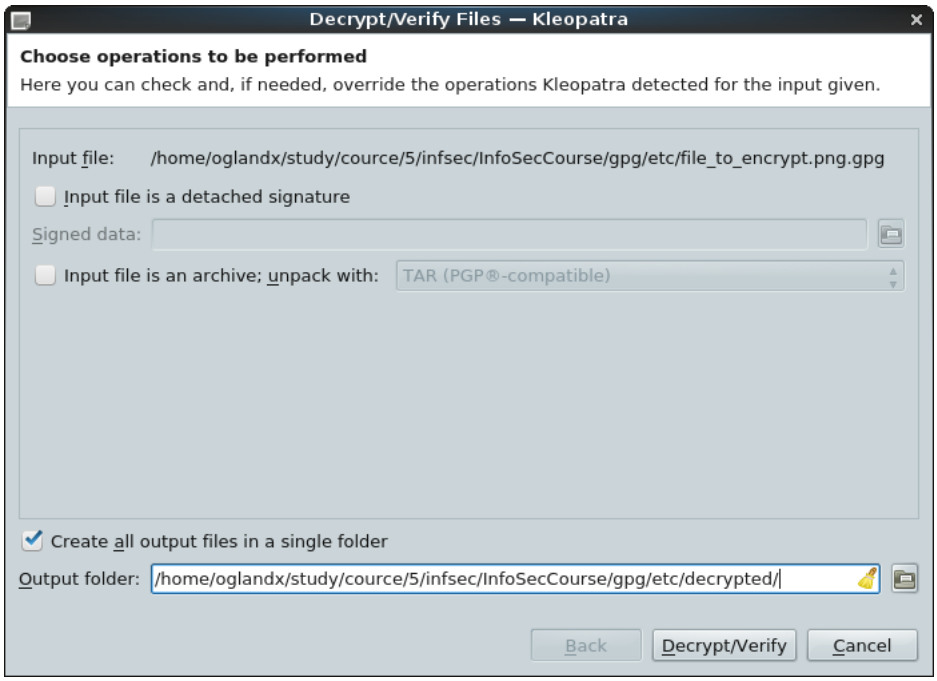

Рис. 19: Дешифрование файла

После нажатия на Decrypt/Verify, файл успешно расшифрован.

4.6 Зашифровать и подписать текст и вместе с сертификатом предоставить коллеге для расшифровки

Создадим файл с текстом для подписи.
Далее откроем диалог Sign/Encrypt выберем Sign and Encrypt.

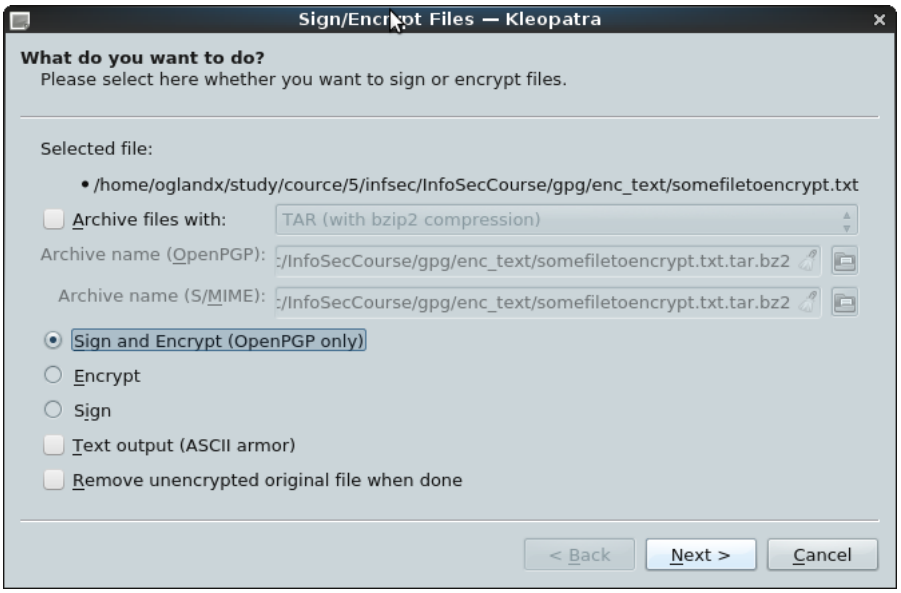

Рис. 20: Выбор Sgin and Encrypt

Выберем сертификат для шифрования.

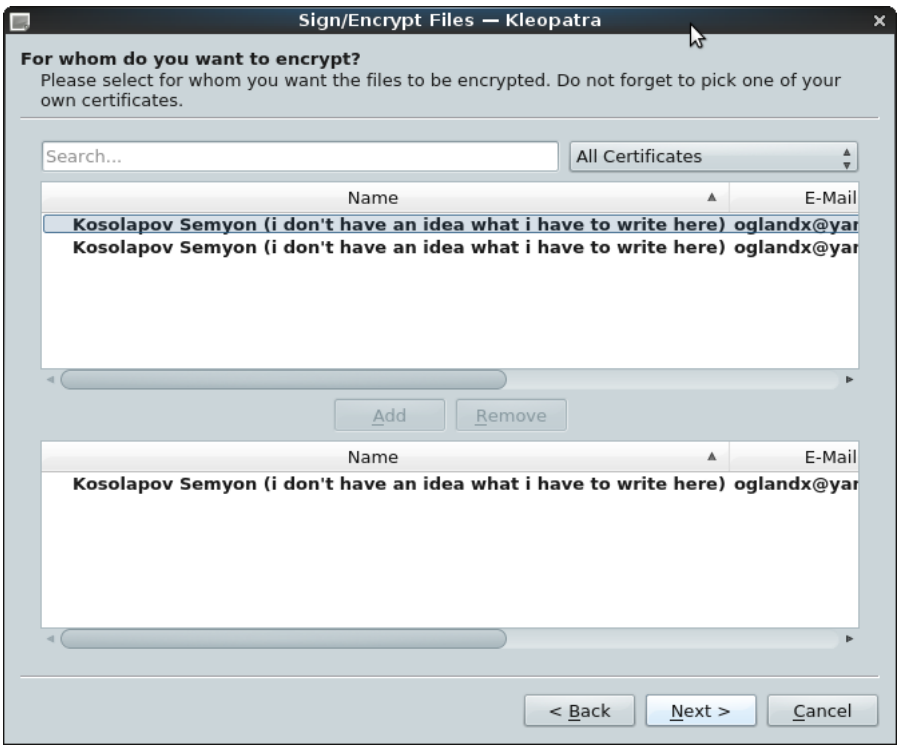

Рис. 21: Выбор сертификата для шифрования

Выберем сертификат для подписи.

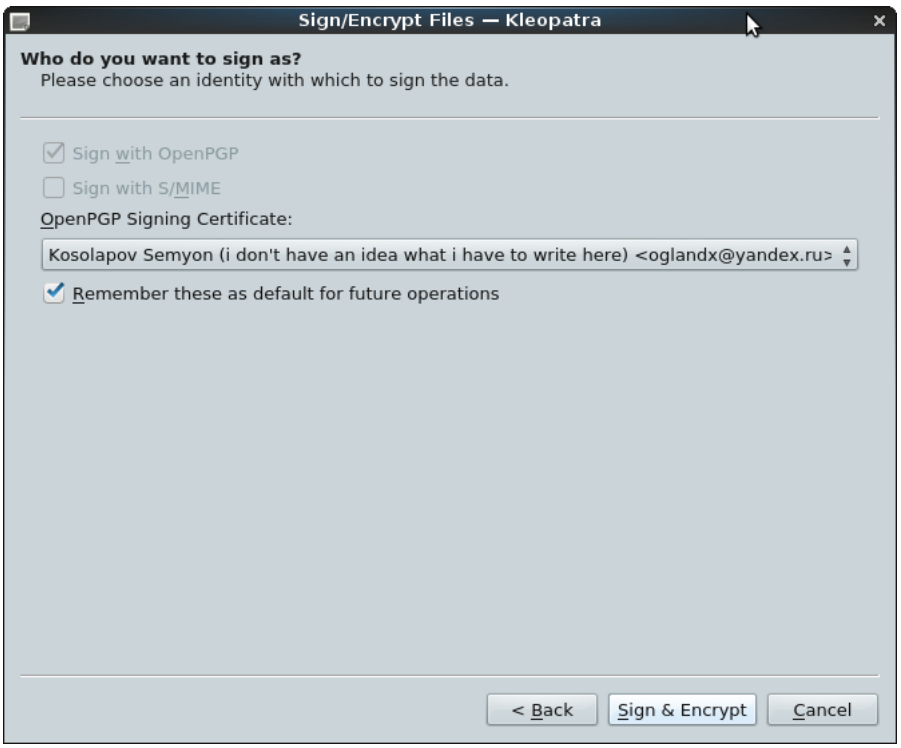

Рис. 22: Выбор сертификата для подписи

В результате - файл успешно подписан и зашифрован.

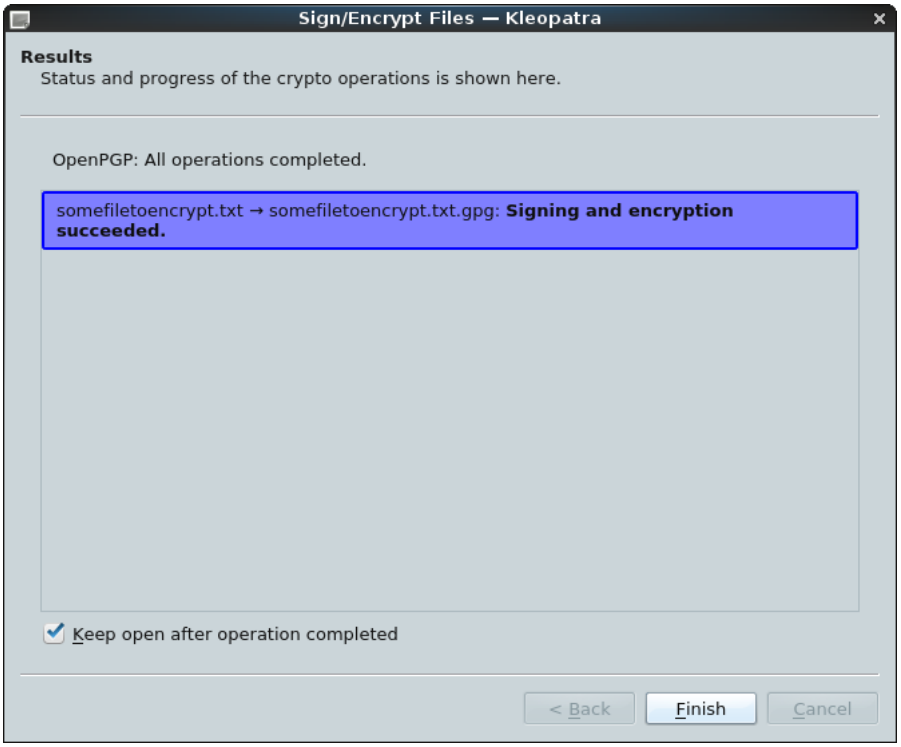

Рис. 23: Операция прошла успешно

4.7 Использование GPG посредством командной строки

Сначала посмотрим, что умеет gpg, выведя help.

```
[oglandx@oglandx console]$ gpg --help
gpg (GnuPG) 2.1.12
```

```
libgcrypt 1.7.0
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Home: ~/.gnupg
Supported algorithms:
Pubkey: RSA, ELG, DSA, ECDH, ECDSA, EDDSA
Cipher: IDEA, 3DES, CAST5, BLOWFISH, AES, AES192, AES256, TWOFISH,
        CAMELLIA128, CAMELLIA192, CAMELLIA256
Hash: SHA1, RIPEMD160, SHA256, SHA384, SHA512, SHA224
Compression: Uncompressed, ZIP, ZLIB, BZIP2

Syntax: gpg [options] [files]
Sign, check, encrypt or decrypt
Default operation depends on the input data

Commands:

 -s, --sign                  make a signature
     --clearsign             make a clear text signature
 -b, --detach-sign           make a detached signature
 -e, --encrypt               encrypt data
 -c, --symmetric             encryption only with symmetric cipher
 -d, --decrypt               decrypt data (default)
     --verify                verify a signature
 -k, --list-keys             list keys
     --list-sigs             list keys and signatures
     --check-sigs            list and check key signatures
     --fingerprint           list keys and fingerprints
 -K, --list-secret-keys      list secret keys
     --gen-key               generate a new key pair
     --quick-gen-key         quickly generate a new key pair
     --quick-adduid          quickly add a new user-id
     --full-gen-key          full featured key pair generation
     --gen-revoke            generate a revocation certificate
     --delete-keys           remove keys from the public keyring
     --delete-secret-keys    remove keys from the secret keyring
     --quick-sign-key        quickly sign a key
     --quick-lsign-key       quickly sign a key locally
     --sign-key              sign a key
     --lsign-key             sign a key locally
     --edit-key              sign or edit a key
     --passwd                change a passphrase
     --export                export keys
     --send-keys             export keys to a key server
     --recv-keys             import keys from a key server
     --search-keys           search for keys on a key server
     --refresh-keys          update all keys from a keyserver
     --import                import/merge keys
     --card-status           print the card status
     --card-edit             change data on a card
     --change-pin            change a card's PIN
     --update-trustdb        update the trust database
     --print-md              print message digests
     --server                run in server mode
     --tofu-policy VALUE     set the TOFU policy for a key

Options:

 -a, --armor                 create ascii armored output
 -r, --recipient USER-ID     encrypt for USER-ID
 -u, --local-user USER-ID    use USER-ID to sign or decrypt
 -z N                        set compress level to N (0 disables)
     --textmode              use canonical text mode
 -o, --output FILE           write output to FILE
 -v, --verbose               verbose
 -n, --dry-run               do not make any changes
 -i, --interactive           prompt before overwriting
     --openpgp               use strict OpenPGP behavior

(See the man page for a complete listing of all commands and options)

Examples:
```

```
 -se -r Bob [file]          sign and encrypt for user Bob
 --clearsign [file]         make a clear text signature
 --detach-sign [file]       make a detached signature
 --list-keys [names]        show keys
 --fingerprint [names]      show fingerprints

Please report bugs to <https://bugs.gnupg.org>.
```

Затем создадим новый сертификат. Создание проходит в интерактивном режиме.

```
[oglandx@oglandx console]$ gpg --gen-key
gpg (GnuPG) 2.1.12; Copyright (C) 2016 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Note: Use "gpg2 --full-gen-key" for a full featured key generation dialog.

GnuPG needs to construct a user ID to identify your key.

Real name: Kosolapov Semyon
Email address: myaddress@gmail.com
You selected this USER-ID:
    "Kosolapov Semyon <myaddress@gmail.com>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
gpg: key C361ED83 marked as ultimately trusted
gpg: revocation certificate stored as '/home/oglandx/.gnupg/openpgp-revocs.d/989
    ↪ F0C2ED5C2BC15778054288FB539FDC361ED83.rev'
public and secret key created and signed.

gpg: checking the trustdb
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: depth: 0  valid:   3  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 3u
pub   rsa2048/C361ED83 2016-06-28 [S]
      Key fingerprint = 989F 0C2E D5C2 BC15 7780  5428 8FB5 39FD C361 ED83
uid         [ultimate] Kosolapov Semyon <myaddress@gmail.com>
sub   rsa2048/2583CFFF 2016-06-28 []
```

Просмотрим имеющиеся подписи и ключи.

```
[oglandx@oglandx console]$ gpg --list-sigs
/home/oglandx/.gnupg/pubring.kbx
------------------------------------
pub   rsa2048/F638203B 2016-06-18 [SC]
uid         [ultimate] Kosolapov Semyon (i don't have an idea what i have to write here) <
    ↪ oglandx@yandex.ru>
sig 3        F638203B 2016-06-18  Kosolapov Semyon (i don't have an idea what i have to write
    ↪ here) <oglandx@yandex.ru>
sub   rsa2048/45941F8D 2016-06-18 [E]
sig          F638203B 2016-06-18  Kosolapov Semyon (i don't have an idea what i have to write
    ↪ here) <oglandx@yandex.ru>

pub   rsa2048/9C7566F3 2016-02-22 [SC]
uid         [ unknown] sba002 <skipalboris@gmail.com>
sig 3        9C7566F3 2016-02-22  sba002 <skipalboris@gmail.com>
sub   rsa2048/00808598 2016-02-22 [E]
sig          9C7566F3 2016-02-22  sba002 <skipalboris@gmail.com>

pub   rsa2048/C361ED83 2016-06-28 [SC]
uid         [ultimate] Kosolapov Semyon <myaddress@gmail.com>
sig 3        C361ED83 2016-06-28  Kosolapov Semyon <myaddress@gmail.com>
sub   rsa2048/2583CFFF 2016-06-28 [E]
sig          C361ED83 2016-06-28  Kosolapov Semyon <myaddress@gmail.com>

[oglandx@oglandx console]$ gpg --list-keys
/home/oglandx/.gnupg/pubring.kbx
------------------------------------
pub   rsa2048/F638203B 2016-06-18 [SC]
```

```
uid          [ultimate] Kosolapov Semyon (i don't have an idea what i have to write here) <
    ↪ oglandx@yandex.ru>
sub   rsa2048/45941F8D 2016-06-18 [E]

pub   rsa2048/9C7566F3 2016-02-22 [SC]
uid          [ unknown] sba002 <skipalboris@gmail.com>
sub   rsa2048/00808598 2016-02-22 [E]

pub   rsa2048/C361ED83 2016-06-28 [SC]
uid          [ultimate] Kosolapov Semyon <myaddress@gmail.com>
sub   rsa2048/2583CFFF 2016-06-28 [E]

[oglandx@oglandx console]$
[oglandx@oglandx console]$ gpg --armor --local-user 2583CFFF --recipient 'Kosolapov Semyon' --
    ↪ encrypt file_to_encrypt.png
[oglandx@oglandx console]$ ls -l
total 120
-rw-r--r-- 1 oglandx oglandx 52785 Jun 28 13:01 file_to_encrypt.png
-rw-r--r-- 1 oglandx oglandx 69089 Jun 28 13:35 file_to_encrypt.png.asc
[oglandx@oglandx console]$ gpg --output decrypted_file.png --decrypt file_to_encrypt.png.asc
gpg: encrypted with 2048-bit RSA key, ID 45941F8D, created 2016-06-18
      "Kosolapov Semyon (i don't have an idea what i have to write here) <oglandx@yandex.ru>"
[oglandx@oglandx console]$
```

Затем зашифруем и расшифруем файл.

```
[oglandx@oglandx console]$
```

Как видим, cmp не имеет вывода, значит файлы идентичны.

5 Выводы

Электронная подпись важных документов и документов, сохранение целостности которых принципиально, необходима. На данный момент она широко используется и, вместе с тем, средства, позволяющие управлять электронной подписью, имеют простой и удобный интерфейс. Причём это касается как графического интерфейса, так и консольного варианта.

GPG и её реализация OpenGPG является одним из вариантов реализации PGP. Оболочка Kleopatra, доступная для ОС семейства Windows и использующих ядро Linux ОС позволяет простыми средствами создавать сертификаты, подписывать и шифровать файлы. При необходимости передачи данных с гарантированной сохранностью целостности и в зашифрованном виде, можно комбинировать эти операции и, зашифровав присланным от человека, которому необходимо передать данные, ключом и последующей установкой подписи, можно гарантировать (хотелось бы в это верить, но всё зависит от алгоритмов подписи и шифрования), что данные будут в сохранности и при этом не станут доступны больше никому.