

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. Тихонова

Департамент электронной инженерии

ОТЧЕТ

О ПРАКТИЧЕСКОЙ РАБОТЕ №4

по дисциплине «Программные и аппаратные средства защиты информации»

«Криптографические ПАСЗИ»

Студент гр. БИБ201

Шадрунов Алексей

Дата выполнения: 21 июня 2023 г.

Преподаватель:

Перов А. А.

« ____ » _____ 2023 г.

Москва, 2023

Содержание

| | | |
|----------|---|-----------|
| 1 | Цель работы | 3 |
| 2 | Ход работы | 3 |
| 2.1 | GPG | 3 |
| 2.1.1 | Создание ключевой пары | 3 |
| 2.1.2 | Просмотр импортированных ключей | 7 |
| 2.1.3 | Экспорт ключей | 8 |
| 2.1.4 | Шифрование файла | 9 |
| 2.1.5 | Расшифрование файла | 10 |
| 2.2 | TrueCrypt | 11 |
| 2.2.1 | Установка | 11 |
| 2.2.2 | Том из файла | 11 |
| 2.2.3 | Шифрование системного раздела | 16 |
| 3 | Выводы о проделанной работе | 21 |

1 Цель работы

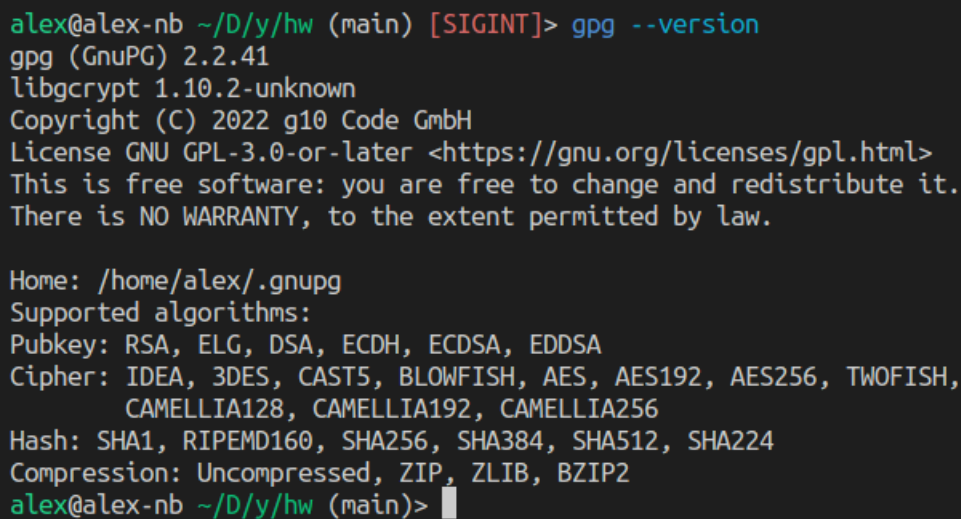
Цель: работа с программно-аппаратными средствами криптографической защиты информации.

2 Ход работы

2.1 GPG

В Linux также есть средства для работы с ключами PGP. На их основе, например, строятся доверительные отношения при распространении ПО: разработчик подписывает дистрибутивы своим ключом, и клиенты могут проверить подлинность файла с помощью опубликованного второго ключа.

В Arch уже установлен пакет gnupg (<https://wiki.archlinux.org/title/GnuPG>) для работы с ключами (Рисунок 1).



```
alex@alex-nb ~/D/y/hw (main) [SIGINT]> gpg --version
gpg (GnuPG) 2.2.41
libgcrypt 1.10.2-unknown
Copyright (C) 2022 g10 Code GmbH
License GNU GPL-3.0-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: /home/alex/.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
alex@alex-nb ~/D/y/hw (main)> █
```

Рисунок 1 – Версия gpg

2.1.1 Создание ключевой пары

Создадим ключевую пару (Рисунок 2). Во время создания утилита просит выбрать тип шифрования, длину ключа, срок действия, имя, адрес и комментарий. После этого появляется приглашение ввести пароль для ключа (Рисунок 3). После этого утилита сообщает, что публичный и секретный ключи созданы и подписаны (Рисунок 4).

```

alex@alex-nb ~/D/y/hw (main)> gpg --full-generate-key
gpg (GnuPG) 2.2.41; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
  (1) RSA and RSA (default)
  (2) DSA and Elgamal
  (3) DSA (sign only)
  (4) RSA (sign only)
 (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072)
Requested keysize is 3072 bits
Please specify how long the key should be valid.
    0 = key does not expire
<n>  = key expires in n days
<n>w  = key expires in n weeks
<n>m  = key expires in n months
<n>y  = key expires in n years
Key is valid for? (0) 365
Key expires at Thu 20 Jun 2024 08:38:57 PM MSK
Is this correct? (y/N) y

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

Real name: Aleksey Shadrinov
Email address: asshadrinov@gmail.com
Comment: test key
You selected this USER-ID:
    "Aleksey Shadrinov (test key) <asshadrinov@gmail.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? 0

```

Рисунок 2 – Версия gpg

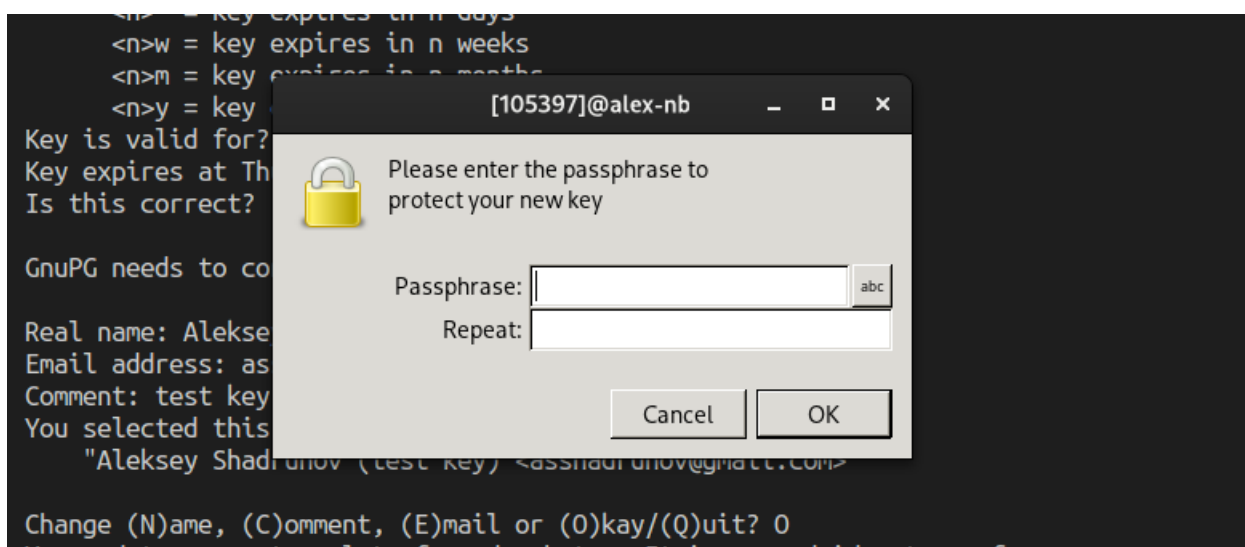


Рисунок 3 – Версия gpg

```

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: directory '/home/alex/.gnupg/openpgp-revocs.d' created
gpg: revocation certificate stored as '/home/alex/.gnupg/openpgp-revocs.d/608EB2AA87
630E97C4442D6D83882AC56F3F549B.rev'
public and secret key created and signed.

pub  rsa3072 2023-06-21 [SC] [expires: 2024-06-20]
    608EB2AA87630E97C4442D6D83882AC56F3F549B
uid                               Aleksey Shadrinov (test key) <asshadrinov@gmail.com>
sub  rsa3072 2023-06-21 [E] [expires: 2024-06-20]

alex@alex-nb ~/D/y/hw (main)>

```

Рисунок 4 – Версия gpg

Можем увидеть созданные ключи в `/.gnupg` (Рисунок 5).

`pubring.kbx` — контейнер с ключами, содержит один или несколько публичных ключей и сертификатов с метаданными. Этот файл позволяет импортировать публичный ключ в другие программы.

Закрытые ключи находятся в папке `private-keys-v1.d`. В папке `openpgp-revocs.d/` находится сертификат для отзыва ключей.

```

alex@alex-nb ~/.gnupg> tree
.
├── crls.d
│   └── DIR.txt
├── openpgp-revocs.d
│   └── 608EB2AA87630E97C4442D6D83882AC56F3F549B.rev
├── private-keys-v1.d
│   ├── 86DCF063C34897159D41EEB9ADC140AEEDBEBFEB.key
│   └── D2999E102FA0CC7CDD6A11676B9994E75B5DB966.key
├── pubring.kbx
├── pubring.kbx~
└── trustdb.gpg

4 directories, 7 files
alex@alex-nb ~/.gnupg>

```

Рисунок 5 – `/.gnupg`

Также ключи можно посмотреть в приложении Passwords and Keys (пакет `seahorse`).

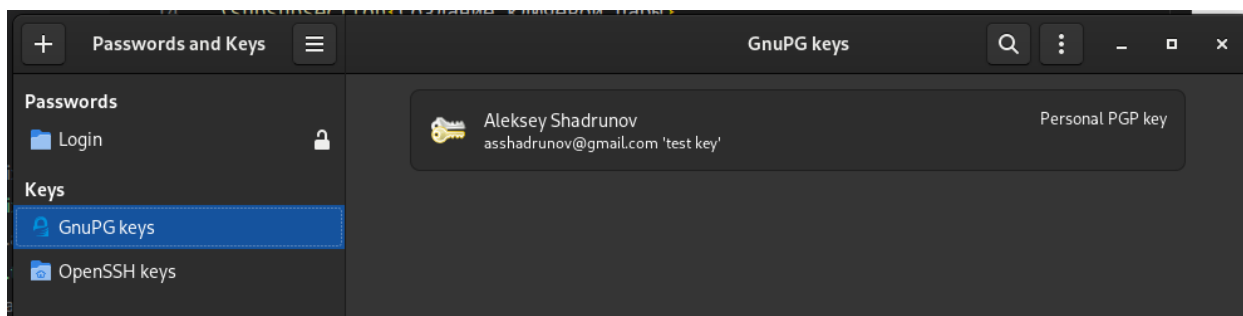


Рисунок 6 – Passwords and Keys

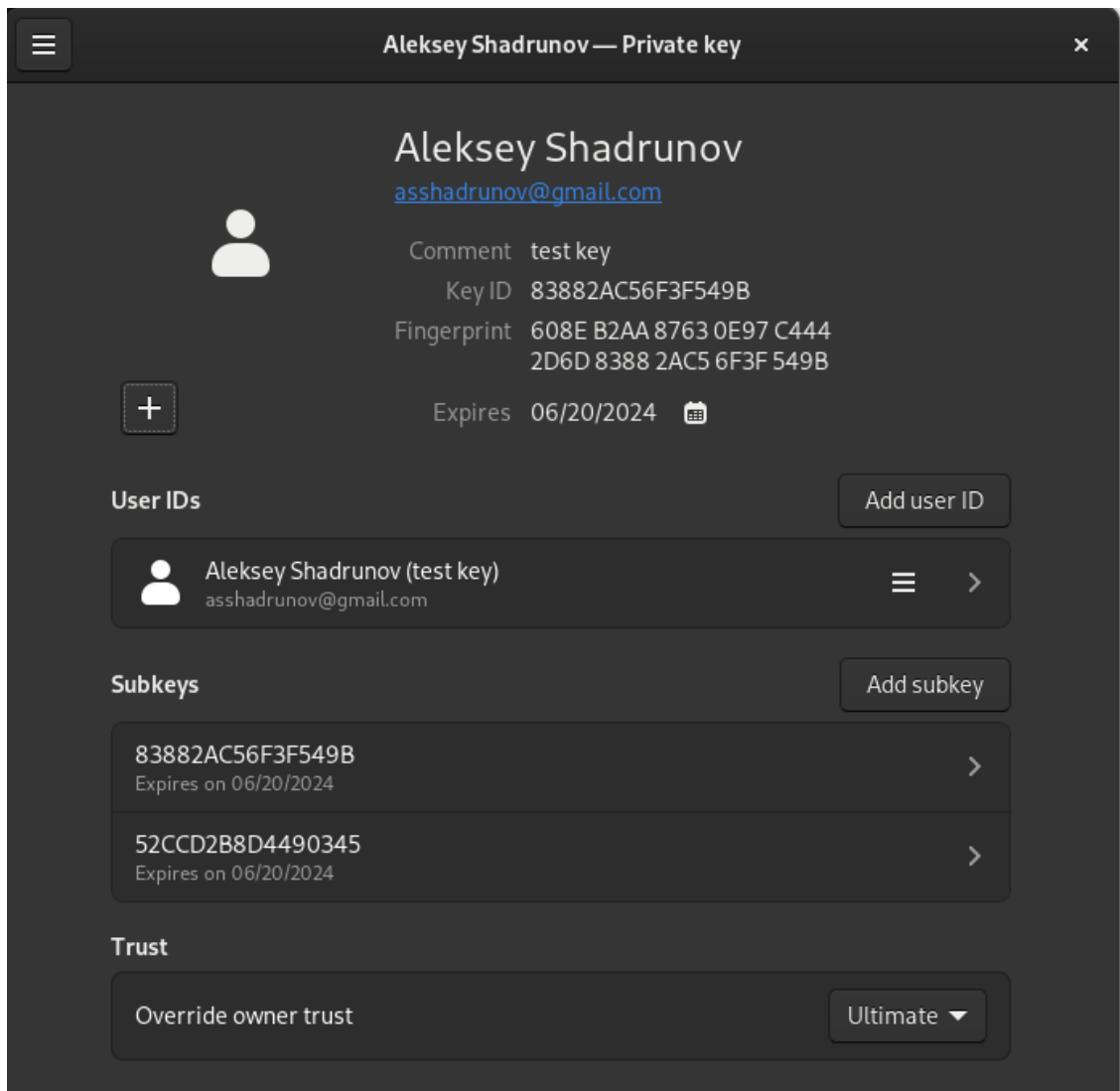


Рисунок 7 – Passwords and Keys

2.1.2 Просмотр импортированных ключей

Для этого выполним команду `gpg --list-keys` (Рисунок 8). Видно, что я импортировал ключ для установки Spotify, для драйвера eToken 5110 от Thales и для чего-то ещё. Также можно посмотреть имеющиеся у меня закрытые ключи командой `gpg --list-secret-keys` (Рисунок 9).

```
alex@alex-nb ~/.g/private-keys-v1.d> gpg --list-keys
/home/alex/.gnupg/pubring.kbx
-----
pub   rsa4096 2021-10-27 [SC] [expired: 2023-01-20]
      F9A211976ED662F00E59361E5E3C45D7B312C643
uid   [ expired] Spotify Public Repository Signing Key <tux@spotify.com>

pub   rsa3072 2020-06-16 [SC]
      B37EBA84D2EB0C786F91EEF77F8AA801285DEE57
uid   [ unknown] Thales DIS <nobody@supportportal.thalesgroup.com>
sub   rsa3072 2020-06-16 [E]

pub   rsa3072 2021-07-30 [SC] [expires: 2024-01-08]
      19882D92DDA4C400C22C0D56CC2AF4472167BE03
uid   [ unknown] Thomas E. Dickey (self-signed w/o SHA1) <dickey@invisible-island.
net>
uid   [ unknown] Thomas E. Dickey (use for email) <dickey@his.com>
sub   rsa3072 2021-07-30 [E] [expires: 2024-01-08]

pub   rsa3072 2023-06-21 [SC] [expires: 2024-06-20]
      608EB2AA87630E97C4442D6D83882AC56F3F549B
uid   [ultimate] Aleksey Shadrunov (test key) <asshadrunov@gmail.com>
sub   rsa3072 2023-06-21 [E] [expires: 2024-06-20]
```

Рисунок 8 – Public keys

```
alex@alex-nb ~/.g/private-keys-v1.d> gpg --list-secret-keys
/home/alex/.gnupg/pubring.kbx
-----
sec   rsa3072 2023-06-21 [SC] [expires: 2024-06-20]
      608EB2AA87630E97C4442D6D83882AC56F3F549B
uid   [ultimate] Aleksey Shadrunov (test key) <asshadrunov@gmail.com>
ssb   rsa3072 2023-06-21 [E] [expires: 2024-06-20]
```

Рисунок 9 – Private keys

2.1.3 Экспорт ключей

Для экспорта выполним команду `gpg --export --armor --output my-key.asc asshadrunov@gmail.com`. Файл появляется в директории (Рисунок 10). Внутри у него записаны байты в стандартной для ключей форме (Рисунок 11).

```
alex@alex-nb ~/D/y/h/4 (main)> gpg --export --armor --output my-key.asc asshadrunov@gmail.com
alex@alex-nb ~/D/y/h/4 (main)> ls -la
total 12
drwxr-xr-x 2 alex alex 4096 Jun 21 20:59 ./
drwxr-xr-x 6 alex alex 4096 Jun 21 20:32 ../
-rw-r--r-- 1 alex alex 2480 Jun 21 20:59 my-key.asc
alex@alex-nb ~/D/y/h/4 (main)>
```

Рисунок 10 – Export key

```
hw > 4 > ≡ my-key.asc
1  -----BEGIN PGP PUBLIC KEY BLOCK-----
2
3  mQGNBGSTNdIBDADjMXRF1XSlTGNk2h1DSLiaAF5S6X6C5v3B1UF69iJzginH0li5
4  1wkjXGuqomz5D7UwyEKny7sT38t4UuE4mCS4F1AqLyBVLiLwv4o1pHbRQnDM26Qs
5  s8mhTwGfc8LnPhgpD9ENQkr7WvHaJXXACC+7Zs1W9cU0vHy+YCDghadQEg7n+NK
6  +2rnIwNNkjsYaa0GliPg60K2NfPjcMEZU5B30NCRE5EpNwkwIKk/2d7wraZlU5ZC
7  W0I9hS9B6HogFWbHgmOQRvrmguQywxN4zGPWpAkB+p21vKvWDl1CfEMGSc7qk0LJ
8  jNYKw0V0jl7Ys8nhziDKY1T8oN7Pp+P84VCFBxHqyPGPiHGNFe7dKXYTIWjFrVx
9  YcV8wYGGPcIkTLQrHqR1xQNu3R9Nr3nTzT5FkEGiSFBCgY6UvWPFVNxUppZ+LXtI
10  5+rHmrCGcjtfLkLFAaEZWvUsmb7+4+kJFEdm7+IxN2p/+A+2oIxM6zh0e/GGM4q
11  VR5JAfDDgYgvFmkAEQEAAbQ0QWxla3NleSBTaGFkcjVub3YgKHRLc3Qga2V5KSA8
12  YXNzaGFkcjVub3ZAZ21haWwY29tPokB1AQTAQgAPhYhBGC0sqqHYw6XxEQtbY0I
13  KsVvP1SbBQJkkzXSAhsDBQkB4TOABQsJCAACBhUKCQgLAQQAQgMBAh4BAheAAAJ
```

Рисунок 11 – Inside asc file

Также утилита позволяет экспортировать ключ на публичные серверы ключей. Такой, например, предоставляет Ubuntu.

2.1.4 Шифрование файла

Для шифрования файла нужно импортировать публичный ключ получателя. Так как мой ключ уже импортирован, зашифруем файл `file.txt` для меня. Для этого выполняю команду `gpg --recipient asshadrinov@gmail.com --encrypt file.txt` (Рисунок 12). Зашифрованный файл появляется в директории (`file.txt.gpg`).

```
alex@alex-nb ~/D/y/h/4 (main)> gpg --recipient asshadrinov@gmail.com --encrypt file.txt
alex@alex-nb ~/D/y/h/4 (main)> ls -la
total 20
drwxr-xr-x 2 alex alex 4096 Jun 21 21:06 ./
drwxr-xr-x 6 alex alex 4096 Jun 21 20:32 ../
-rw-r--r-- 1 alex alex  12 Jun 21 21:04 file.txt
-rw-r--r-- 1 alex alex  478 Jun 21 21:06 file.txt.gpg
-rw-r--r-- 1 alex alex 2480 Jun 21 20:59 my-key.asc
alex@alex-nb ~/D/y/h/4 (main)>
```

Рисунок 12 – Encrypted file

Можем зашифровать файл в ASCII-виде с помощью опции `--armor` (Рисунок 13). Тогда его зашифрованное содержимое легко посмотреть и передать сообщением (Рисунок 14).

```
alex@alex-nb ~/D/y/h/4 (main) [2]> gpg --recipient asshadrinov@gmail.com --armor --encrypt file.txt
alex@alex-nb ~/D/y/h/4 (main)> ls -la
total 24
drwxr-xr-x 2 alex alex 4096 Jun 21 21:10 ./
drwxr-xr-x 6 alex alex 4096 Jun 21 20:32 ../
-rw-r--r-- 1 alex alex  12 Jun 21 21:04 file.txt
-rw-r--r-- 1 alex alex  711 Jun 21 21:10 file.txt.asc
-rw-r--r-- 1 alex alex  478 Jun 21 21:06 file.txt.gpg
-rw-r--r-- 1 alex alex 2480 Jun 21 20:59 my-key.asc
alex@alex-nb ~/D/y/h/4 (main)>
```

Рисунок 13 – ASCII file

```
hw > 4 > file.txt.asc
1  -----BEGIN PGP MESSAGE-----
2
3  hQGMA1LM0rjUSQNFAQv+Pew0ogC8pREZW/XeEMUf9t2lLsvDcKbMAsMNqpXOLifs
4  0LHt2ZPEGpvsz4LDDB7h8FDH1/pRyLWfdbnyLZv0EAKhmgAna70r9yBZEriarbrw
5  FvXjYvQ9fnm2mgvSlmA+txdm8CxXN0hSZ66dPRuJzpbLTNXHSqLwnySKF4wvbupp
6  rQsf6hKQ7AY32f9tF7p6U5D/G8JgCf+7Aj3/hVVYpQyZ5bS5kYsEYsKF5s/5Ubr
7  DJhSV52R0/H/7Dg6iFdf7fxiQzMdMeHa8mM/H2lNIzVc5IU/TyIoRDynhzzP9MtI
8  8Hn0jqwqBMBwABR4srC+aPltRFIPJ+D3IzwuvsLwi8/VswZu4D3UNltuWFeY6eL0
9  vrGbtDvug9A/PMM5ZiA0pk06+z5HX2Kp54m1/oIPrIHJW4/HMmPYwFavxuN1kR93
10  3cbkqxcdk0V3DBX1mfEjICFREugH1y/edhdwD05uIp2BT+afLcv5yBSNhAtHKAbt
11  kwXQ4If1HZRffz9HtJ1l0k0BIE/QIQX+9wGEZyZtRz0/LvbF2JcoDizGxwj2rSv4
12  HgrJmLKia1XI0ot3WyJI3LFV6nH65rohWn0FAz4bWEr4wjVv+k0ftxpP9W07XA==
13  =uhuF
14  -----END PGP MESSAGE-----
15
```

Рисунок 14 – ASCII file

2.1.5 Расшифрование файла

Расшифрование производится командой `gpg --decrypt file.txt.asc` (Рисунок 12). Приглашение попросит ввести парольную фразу от ключа.

```
alex@alex-nb ~/D/y/h/4 (main)> gpg --decrypt file.txt.gpg
gpg: encrypted with 3072-bit RSA key, ID 52CCD2B8D4490345, created 2023-06-21
      "Aleksey Shadrinov (test key) <asshadrinov@gmail.com>"
my new file
alex@alex-nb ~/D/y/h/4 (main)> gpg --decrypt file.txt.asc
gpg: encrypted with 3072-bit RSA key, ID 52CCD2B8D4490345, created 2023-06-21
      "Aleksey Shadrinov (test key) <asshadrinov@gmail.com>"
my new file
alex@alex-nb ~/D/y/h/4 (main)> █
```

Рисунок 15 – Decrypted file

2.2 TrueCrypt

2.2.1 Установка

Установим TrueCrypt на виртуальную машину (Рисунок 16).

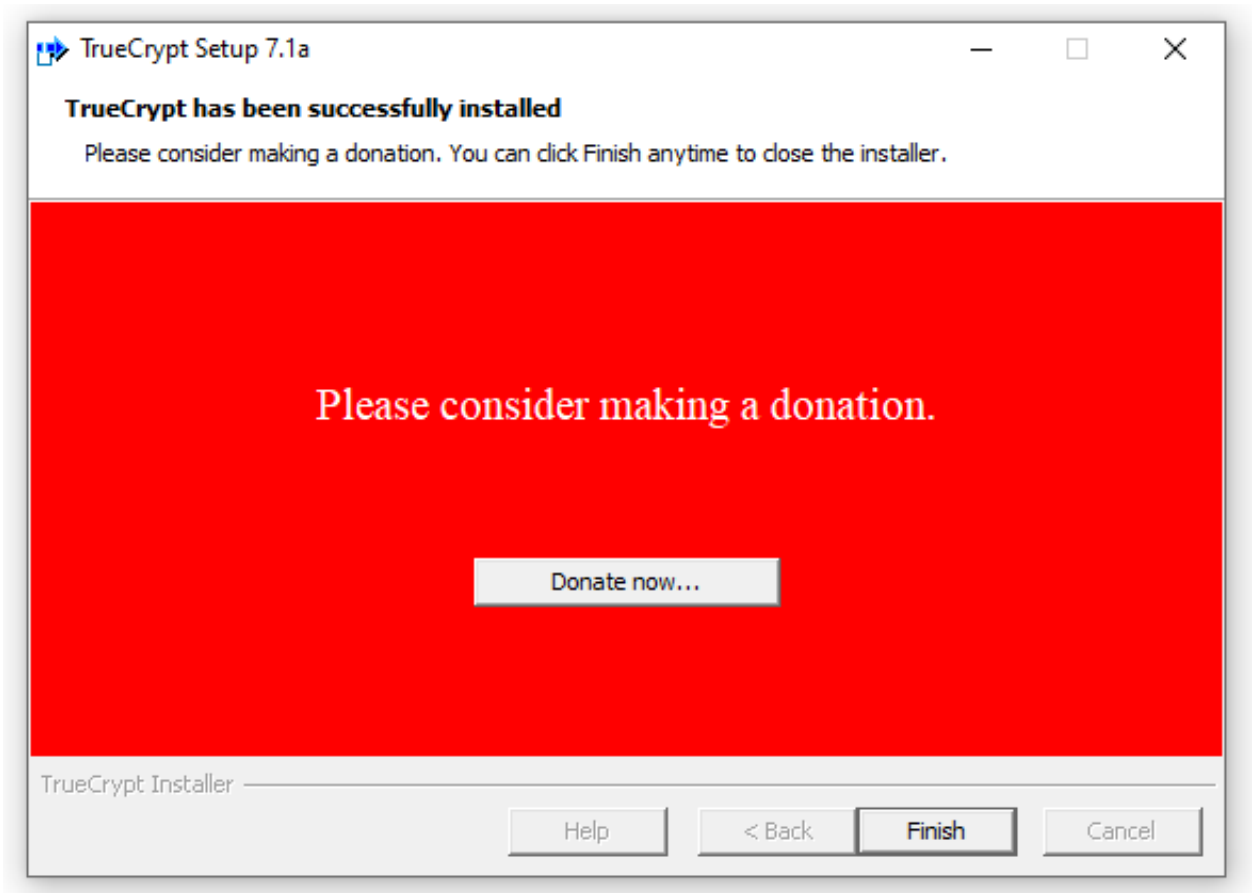


Рисунок 16 – Installation

2.2.2 Том из файла

Создадим том TrueCrypt. Для этого откроем окно программы и выберем Create Volume (Рисунок 17).

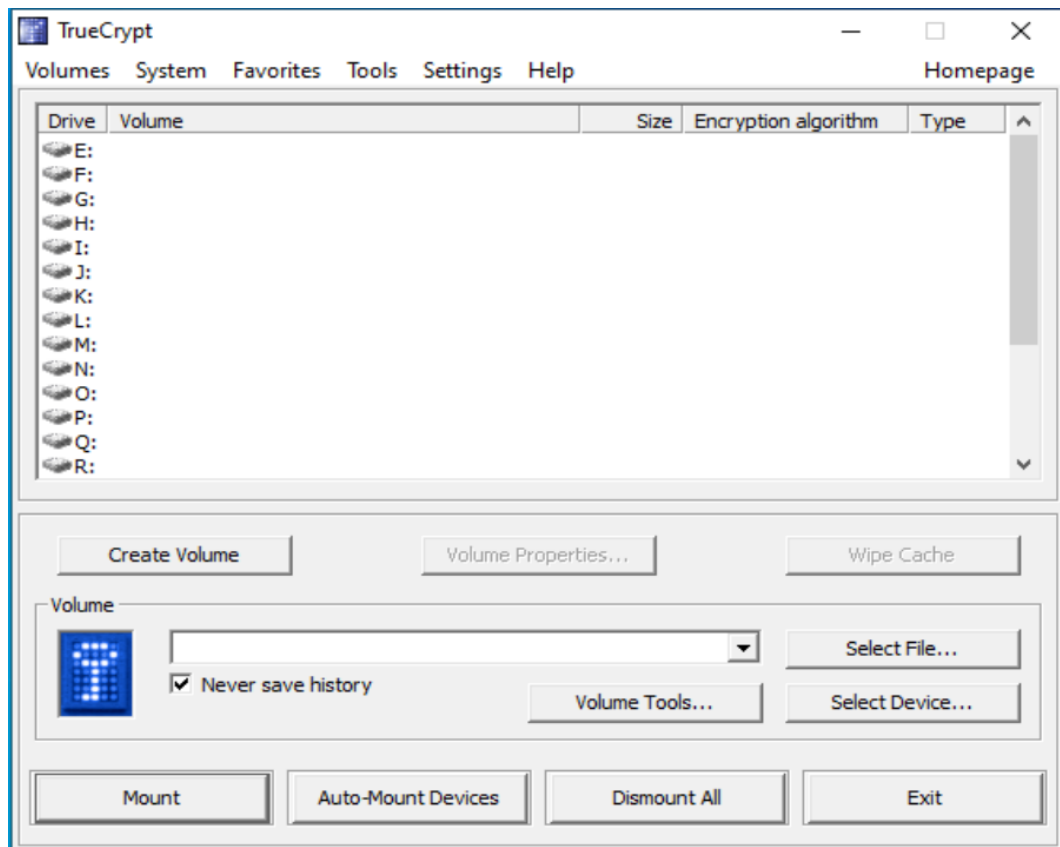


Рисунок 17 – TrueCrypt

На следующем шаге выбираем файловый контейнер, выбираем расположение, размер тома, задаём пароль (Рисунки 18-22).

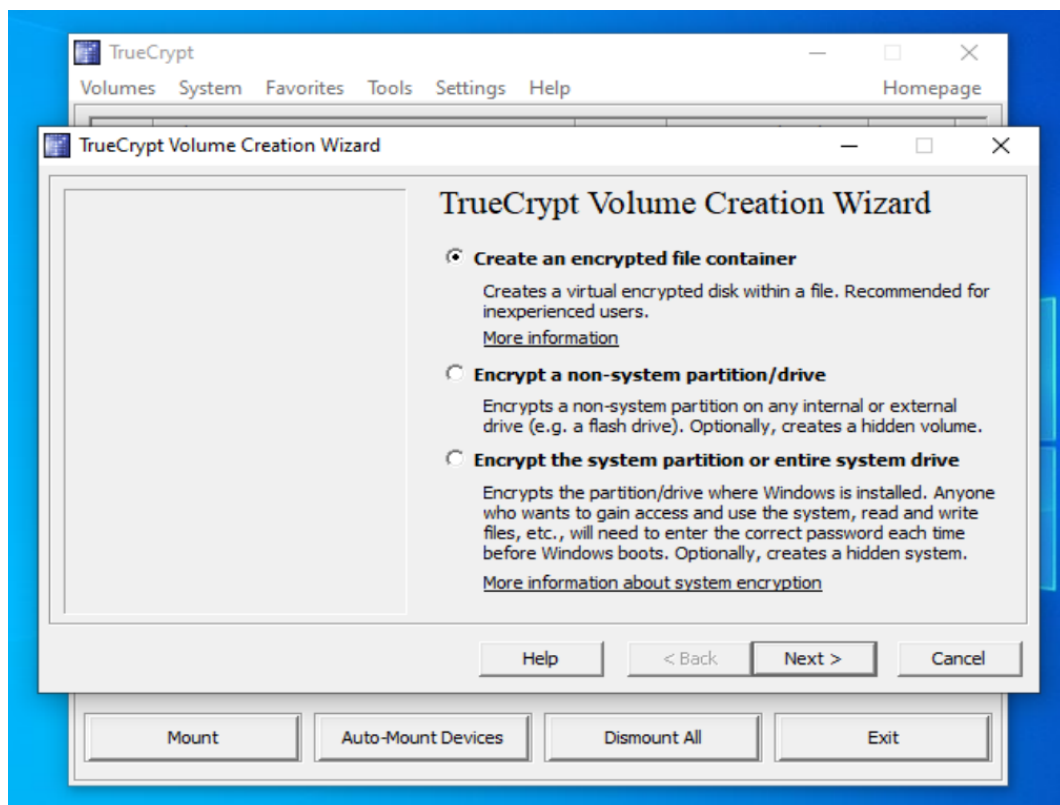


Рисунок 18 – Create container

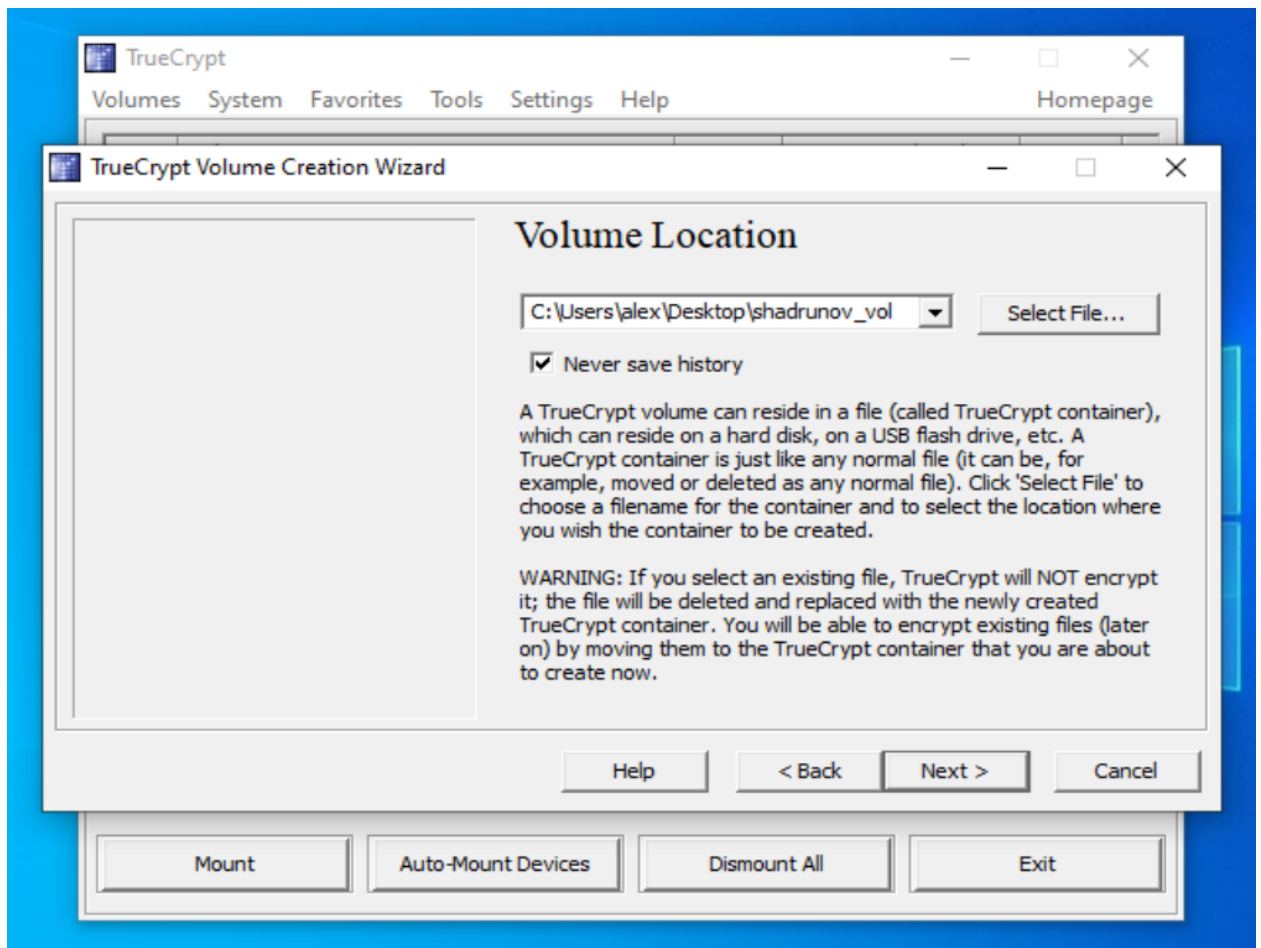


Рисунок 19 – Location

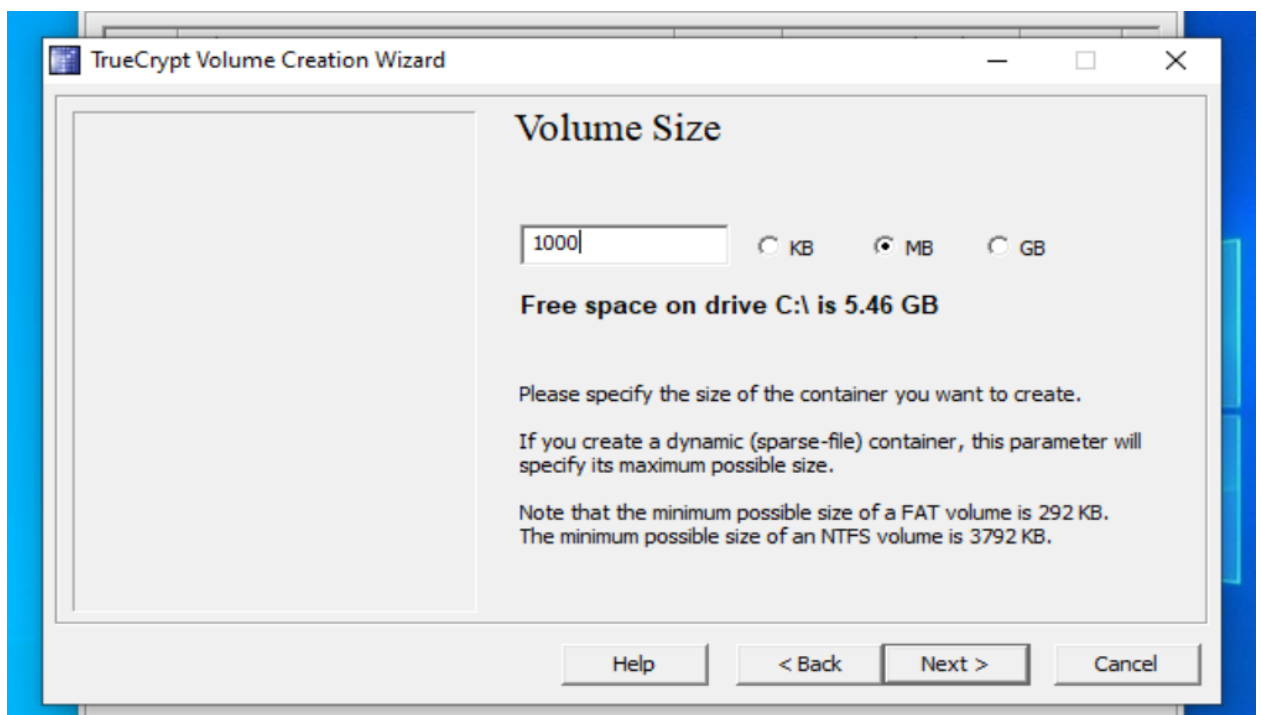


Рисунок 20 – Volume size

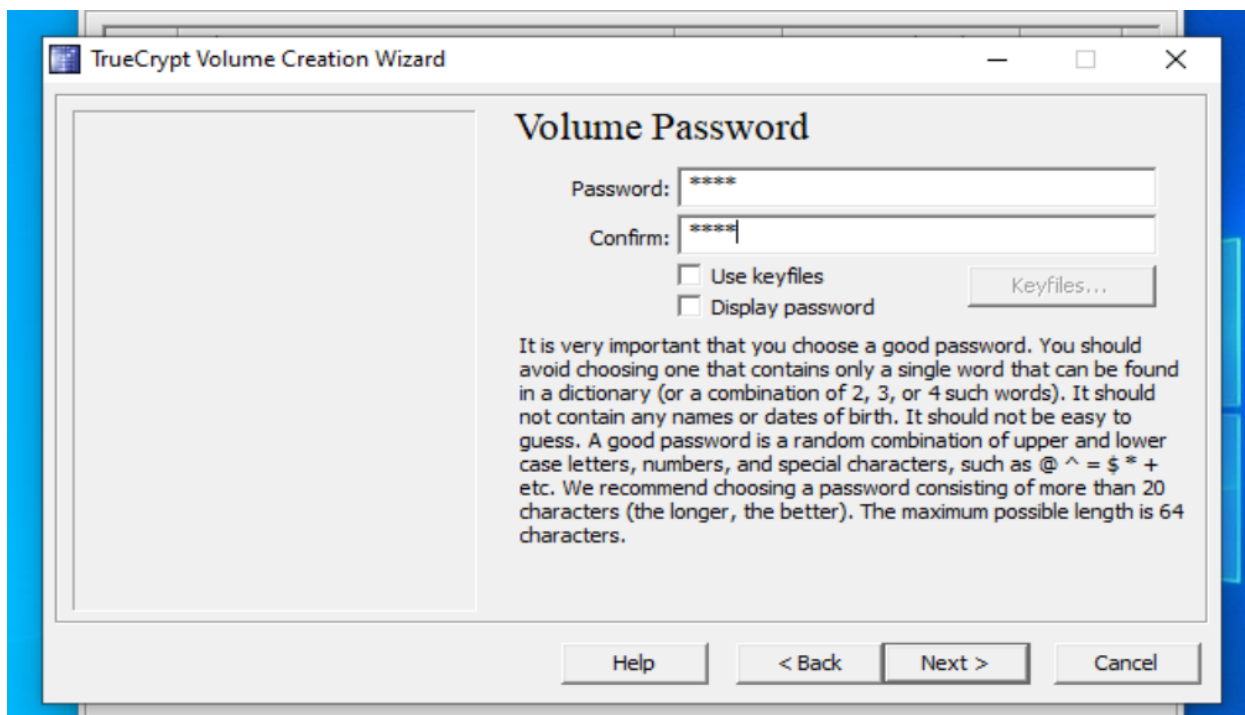


Рисунок 21 – Password

Теперь мы можем нажать Select File, а затем Mount, после чего том смонтируется к диску Q: (Рисунок 22). При этом этот диск отображается в системе (в Проводнике).

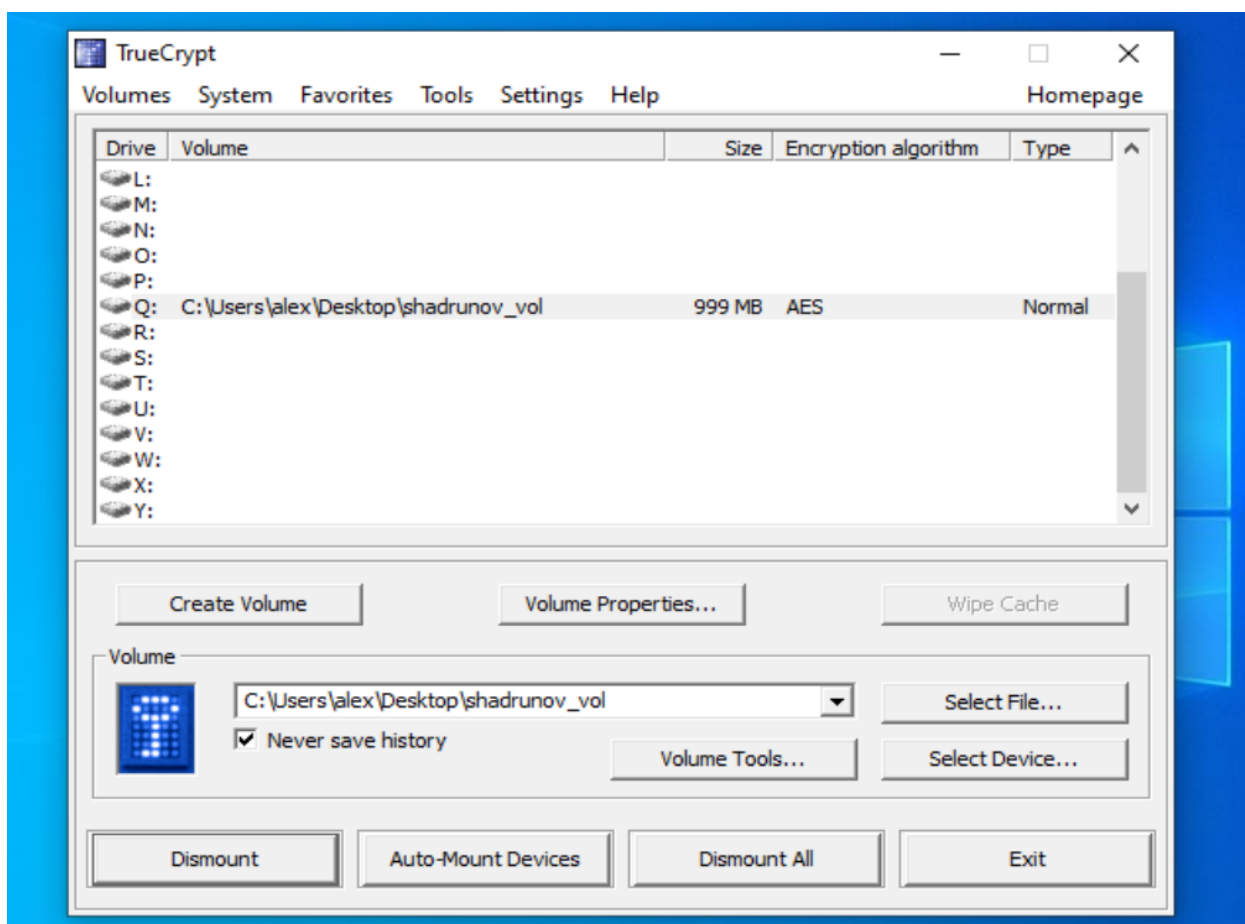


Рисунок 22 – Mounted volume

Если нажать Dismount, всё вернётся как было (Рисунок 23).

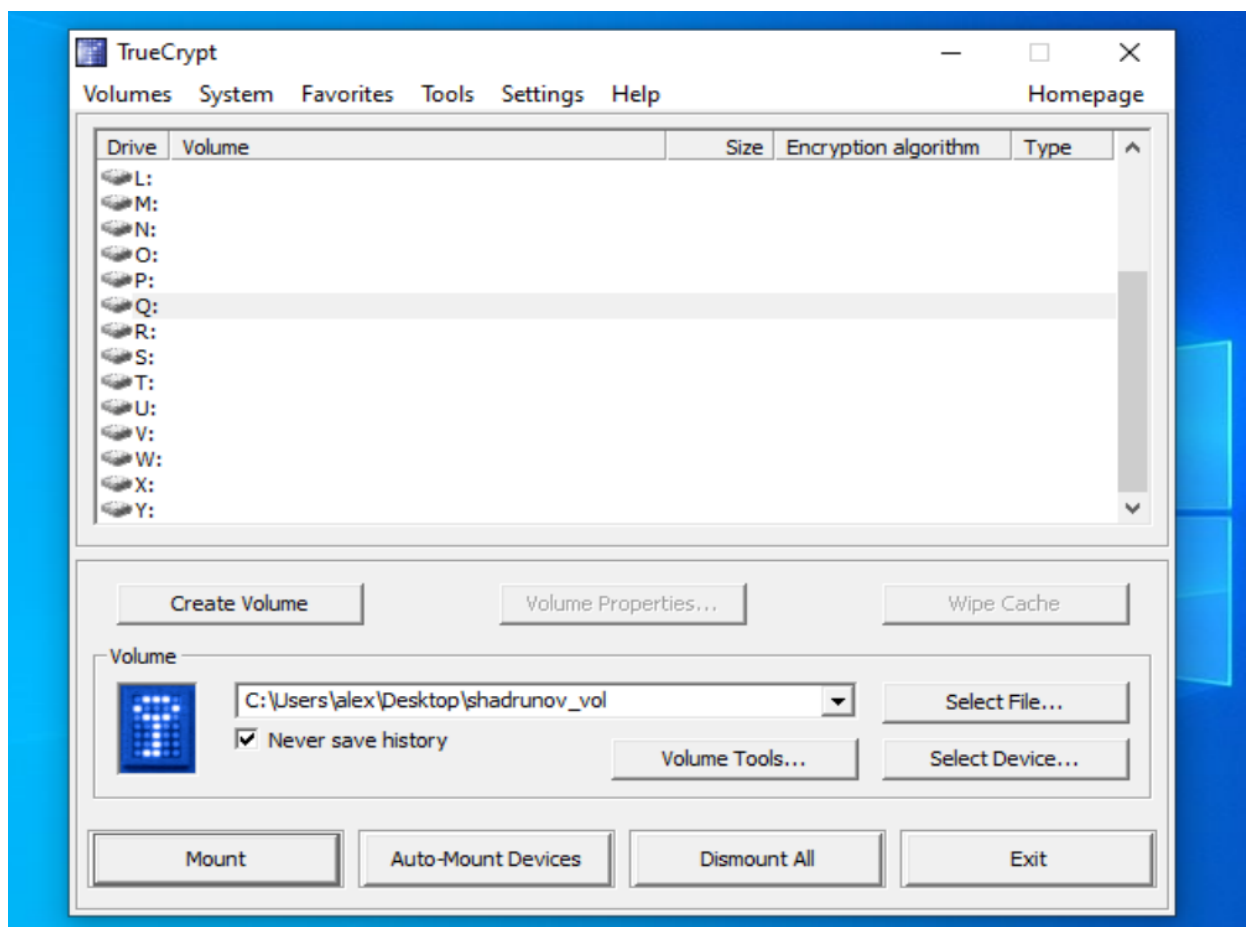


Рисунок 23 – Dismount

2.2.3 Шифрование системного раздела

Запустим System > Encrypt system partition. Далее выбираем область для шифрования (только раздел с Windows), опции шифрования, пароль. После этого мастер сгенерирует ключи и попросит записать их на диск. Это можно пропустить, если запустить мастера с ключом /п. После этого система предложит перезагрузить компьютер (Рисунки 24-28).

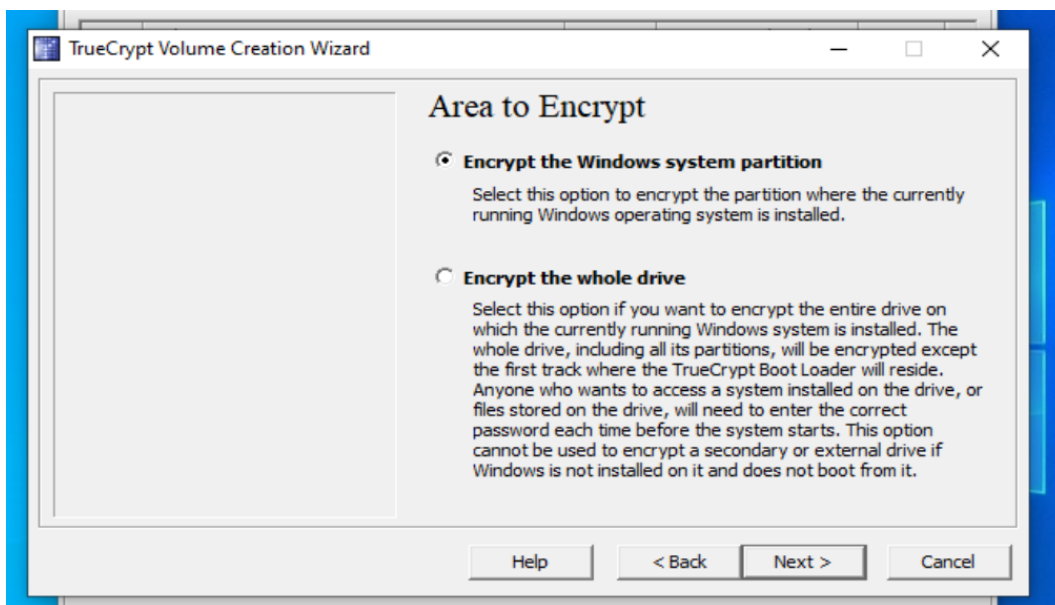


Рисунок 24 – Encrypt system partition



Рисунок 25 – Encryption options

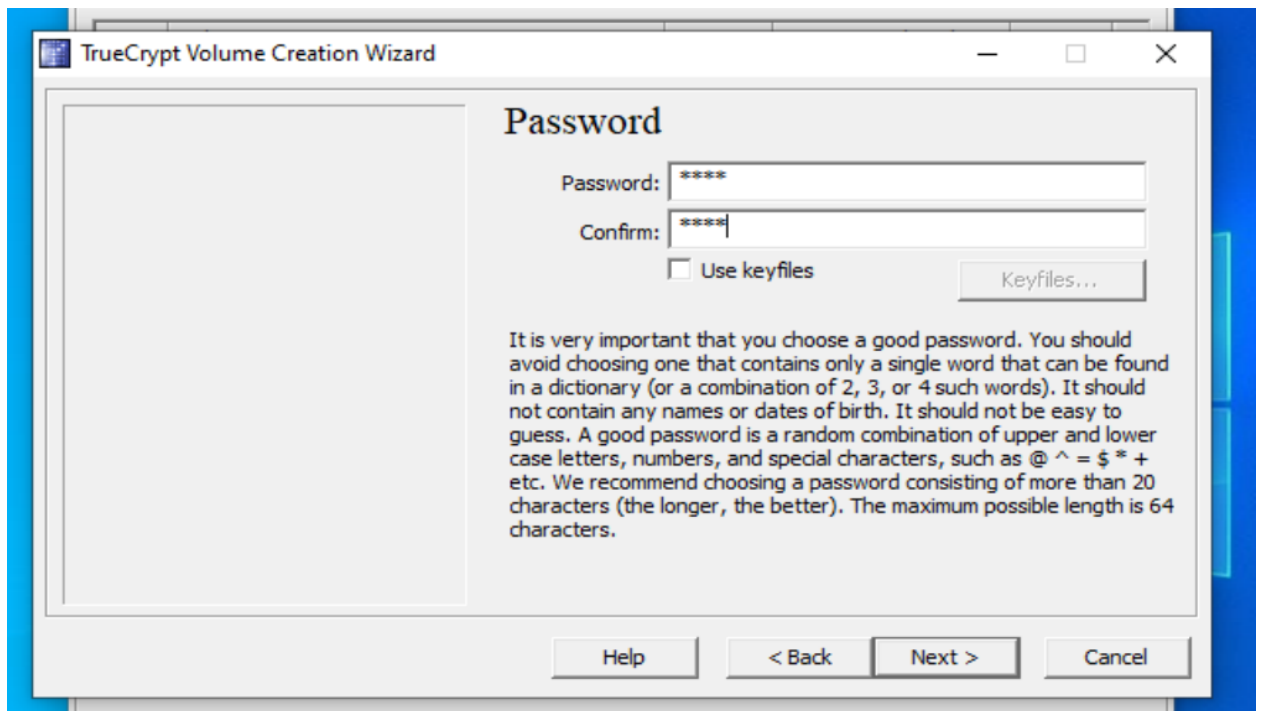


Рисунок 26 – Password

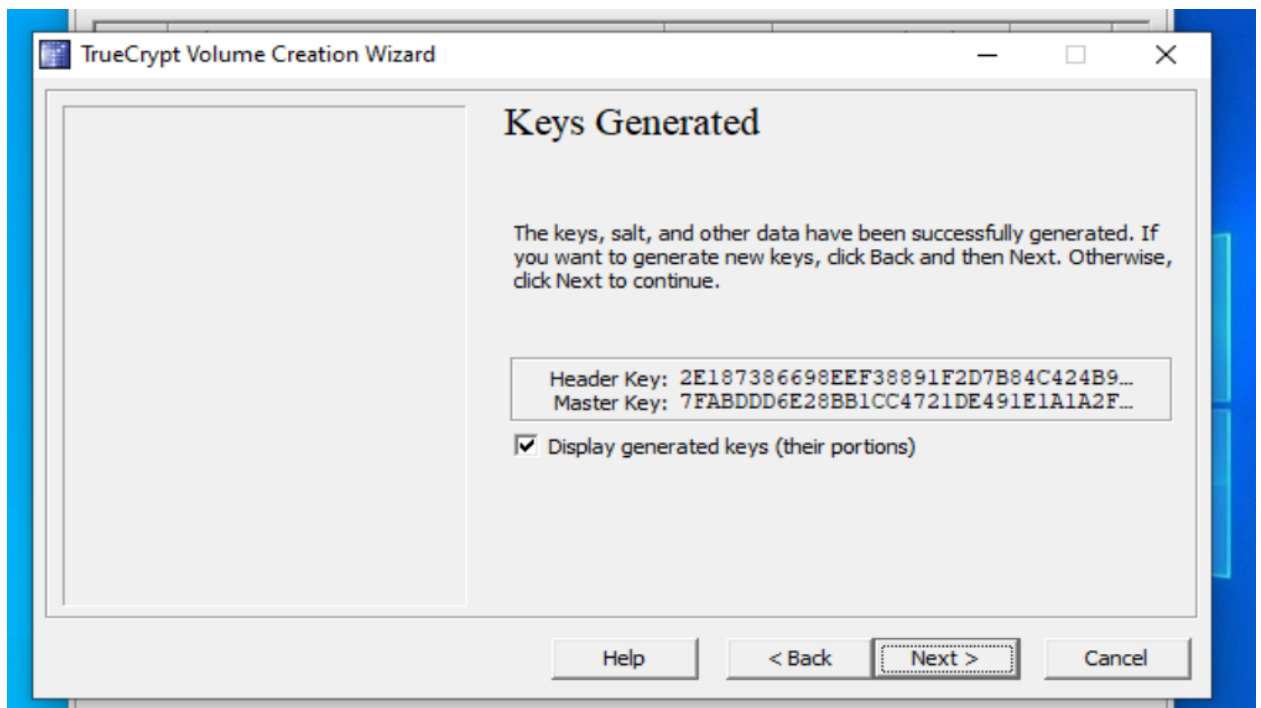


Рисунок 27 – Keys generated

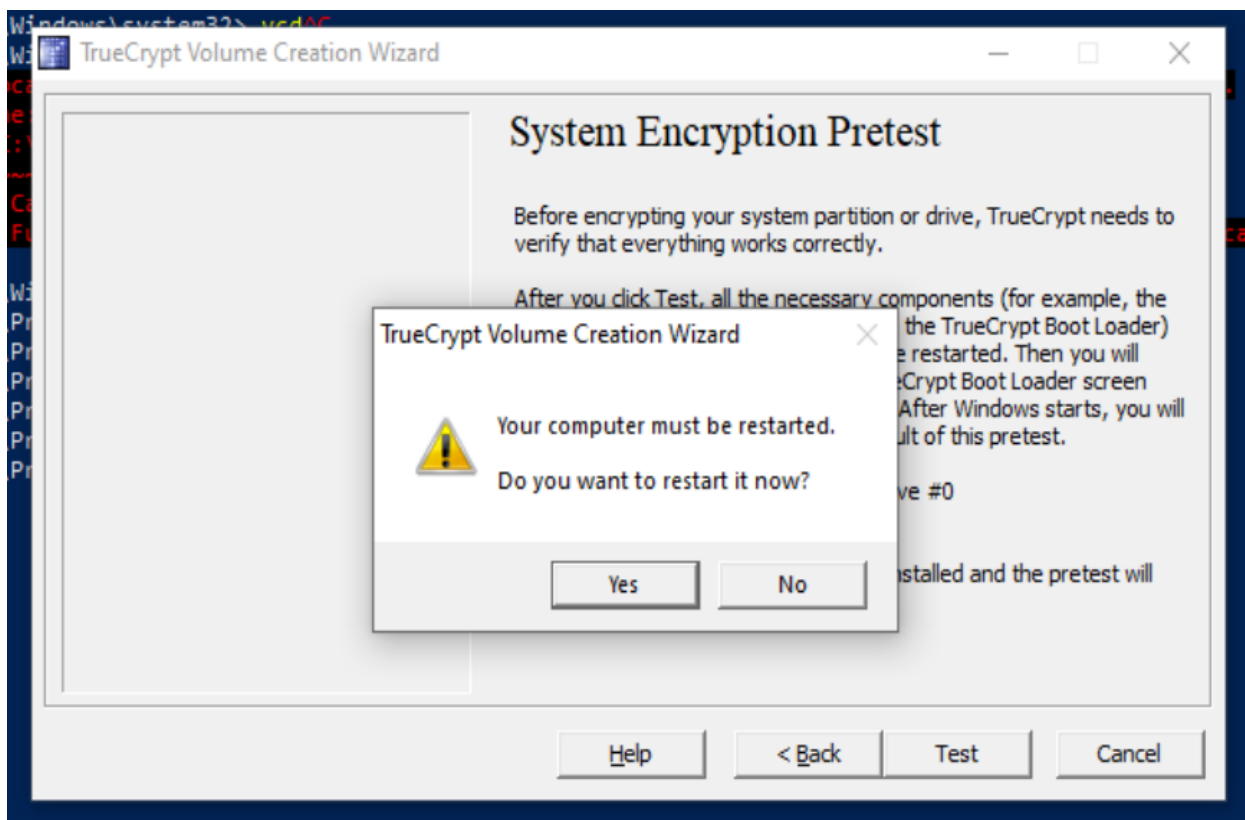


Рисунок 28 – Prompt to reboot

При перезагрузке вместо загрузчика Windows мы увидим TrueCrypt Boot Loader, который просит ввести пароль, чтобы расшифровать раздел с Windows (Рисунок 29).



Рисунок 29 – TrueCrypt Boot Loader

После успешного ввода пароля загрузка продолжается (Рисунок 30), а мастер сообщает нам, что проверка пройдена (Рисунок 31).

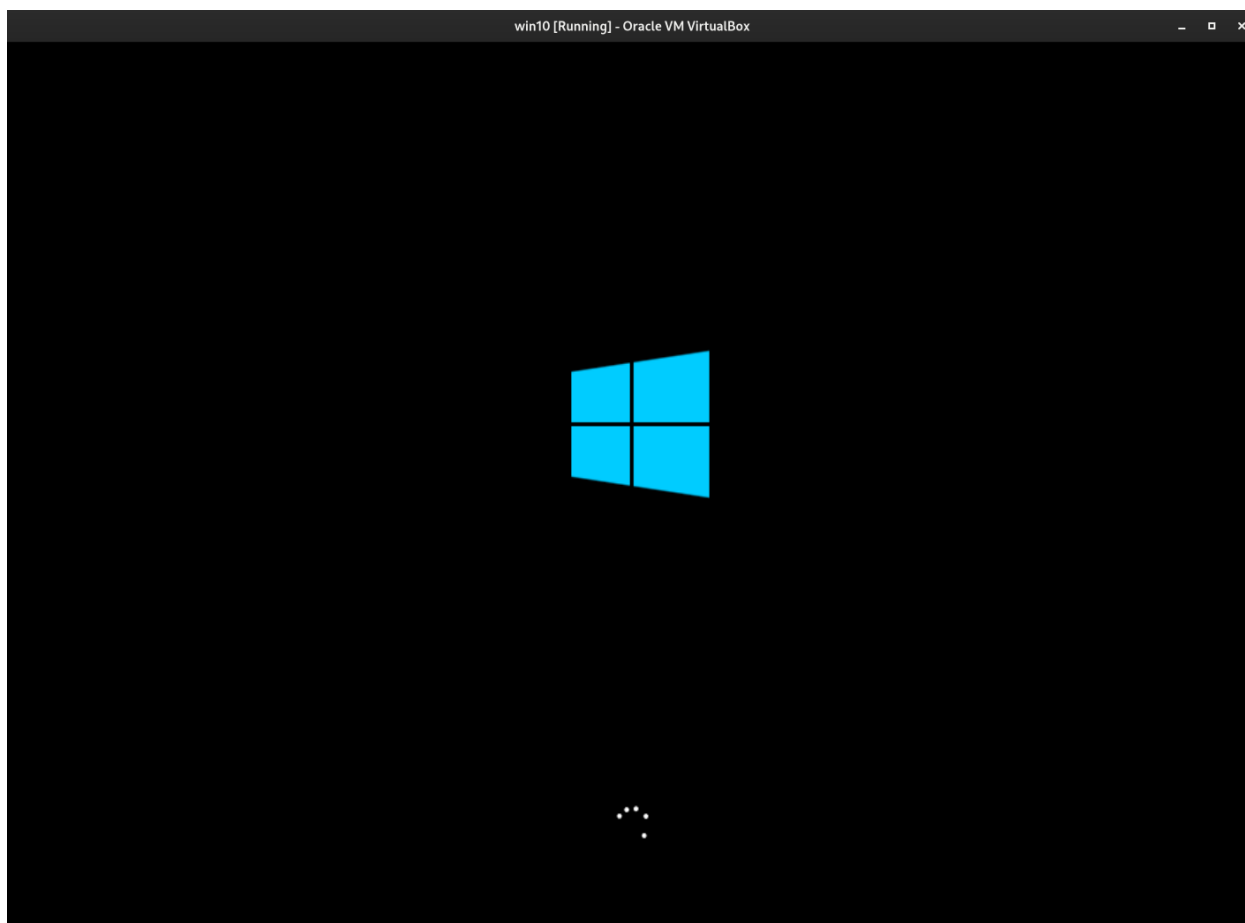


Рисунок 30 – Windows loading

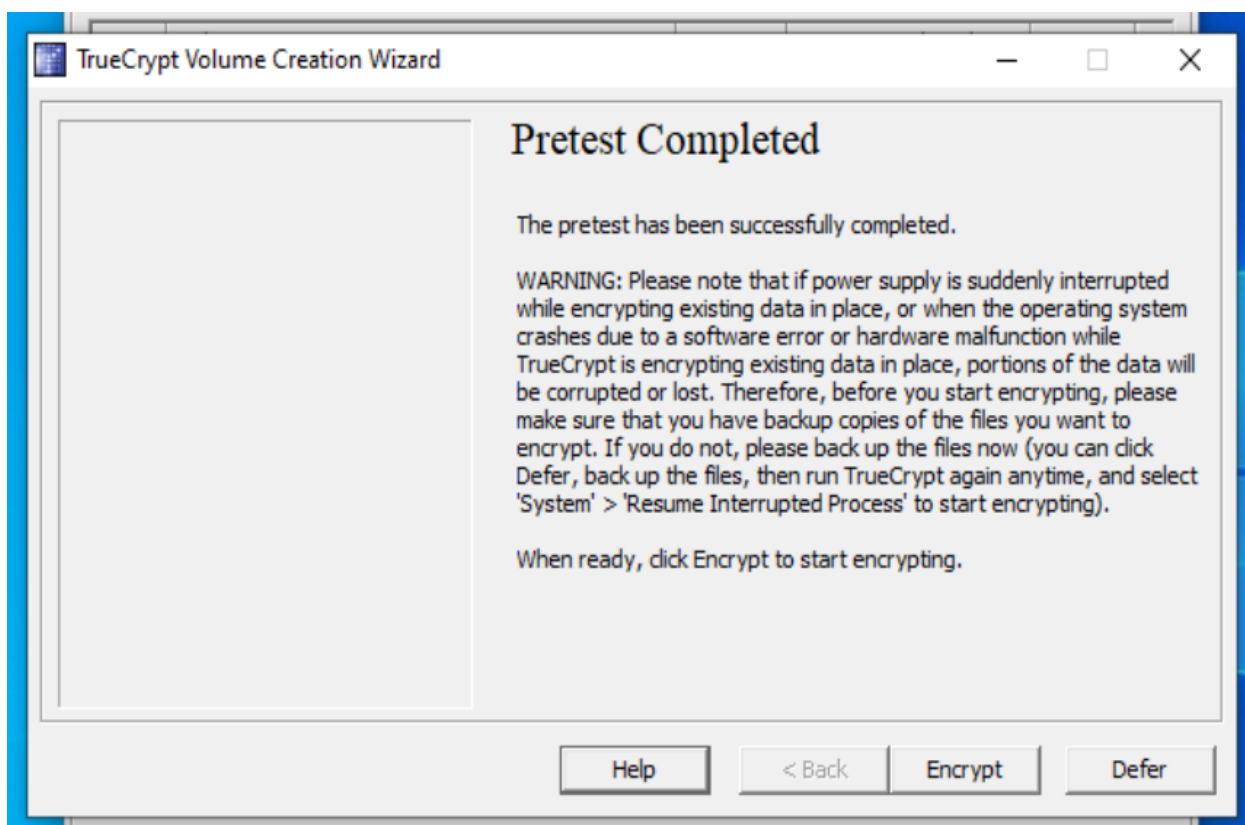


Рисунок 31 – Pretest completed

После этого начинается шифрование диска. Прогресс виден на рисунке 32.

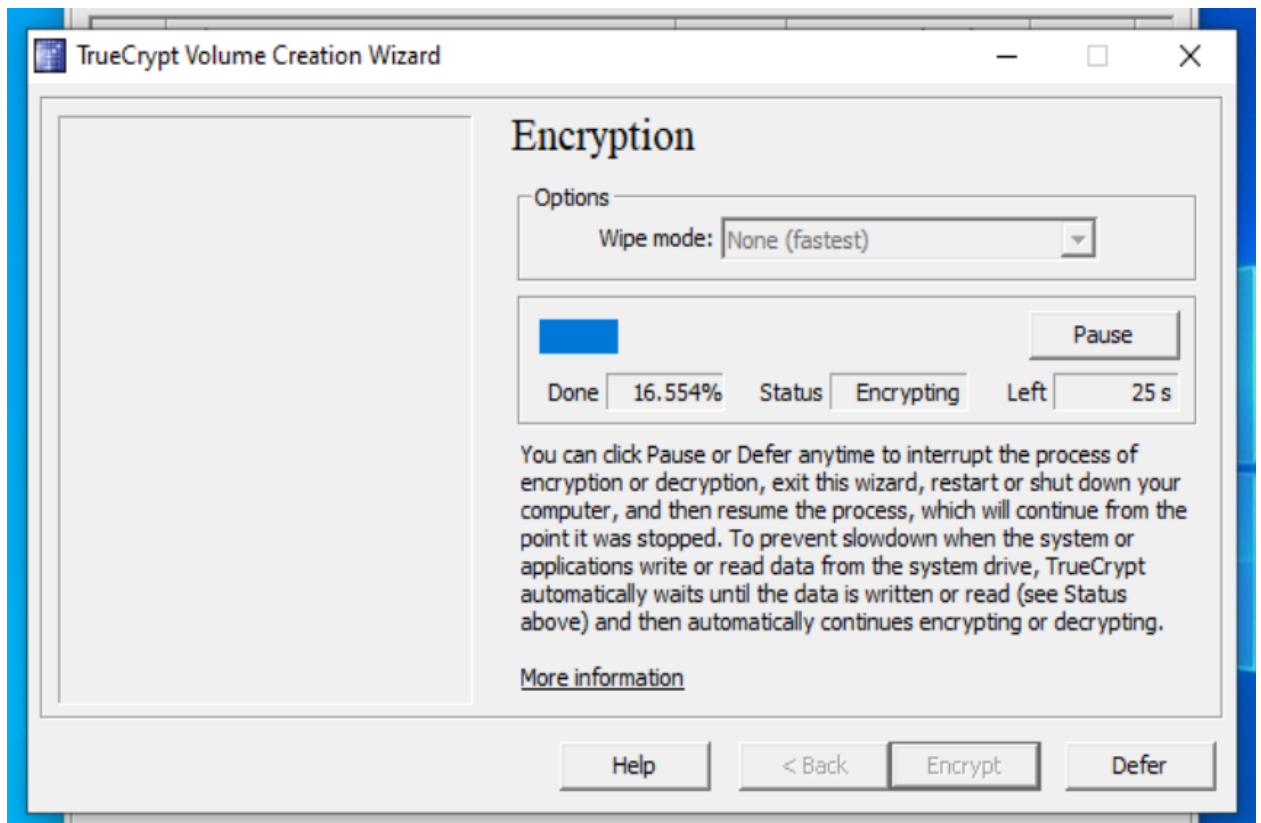


Рисунок 32 – Disk encryption

После этого раздел С: отображается в TrueCrypt как зашифрованный том (Рисунок 33).

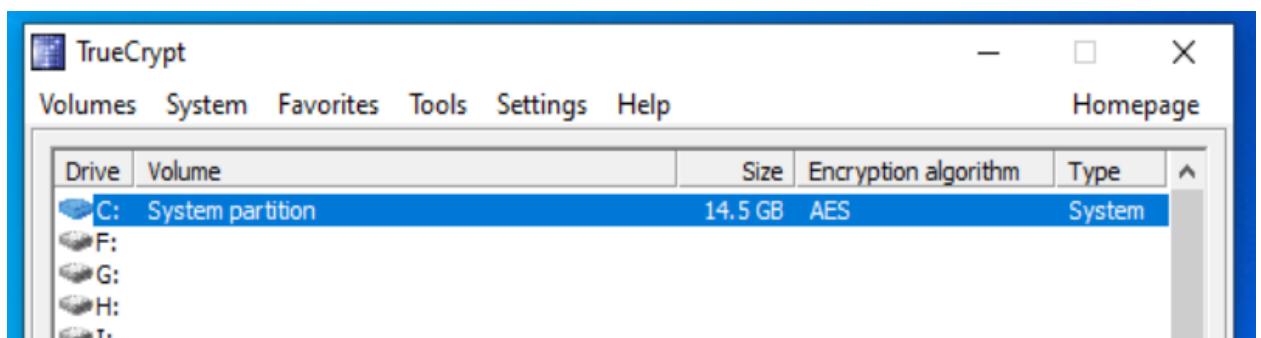


Рисунок 33 – System partition in TrueCrypt

3 Выводы о проделанной работе

Я изучил работу с программно-аппаратными средствами криптографической защиты информации на примере программы для шифрования grg и программы для шифрования томов TrueCrypt.