



[Главная](#) >> [Инструкции](#) >> Шифрование файлов и папок в Linux

## Шифрование файлов и папок в Linux

Обновлено: 07 мая 2020 Опубликовано: 15 августа, 2015 от [admin](#), 5 комментариев, время чтения: 7 минут

Обнаружили ошибку в тексте? Сообщите мне об этом. Выделите текст с ошибкой и нажмите **Ctrl+Enter**.

В современном мире каждый аспект нашей личной жизни записывается на компьютеры. Один из способов защиты наиболее важной информации - шифрование файлов и каталогов. Во время шифрования содержимое файлов перемешивается с избыточными данными в соответствии с установленным алгоритмом, таким образом, что расшифровать его можно только имея специальный пароль или ключ.

В операционной системе Linux есть замечательный инструмент с открытым исходным кодом для шифрования файлов - GNU Privacy Guard или просто GPG, который может быть использован для шифрования любого файла из командной строки или в графическом режиме. О нем и пойдет речь в сегодняшней статье.

### Содержание статьи

- [Утилита GPG](#)
- [Шифрование файлов с помощью пароля](#)
- [Шифрование с использованием ключей](#)

Конфиденциальность - [Условия использования](#)

Privacy

- [Подписи и шифрование](#)
- [Выводы](#)

## Утилита GPG

Перед тем как перейти к использованию утилиты, давайте рассмотрим ее синтаксис:

**\$ gpg опции файл параметры**

Опции указывает что необходимо сделать с файлом, как это сделать и какие возможности использовать. Давайте рассмотрим самые основные опции, которые мы будем использовать в этой статье:

- **-h** - вывести справку по утилите;
- **-s, --sign** - создать цифровую подпись, эта опция используется вместе с другими опциями для шифрования;
- **--clearsign** - подписать незашифрованный текст;
- **-e, --encrypt** - зашифровать данные, с помощью ключа;
- **-c, --symmetric** - зашифровать данные, с помощью пароля;
- **-d, --decrypt** - расшифровать данные, зашифрованные с помощью ключа или пароля;
- **--verify** - проверить подпись;
- **-k, --list-keys** - вывести доступные ключи;
- **--list-sigs** - вывести доступные подписи;
- **--fingerprint** - вывести все ключи вместе с их отпечатками;
- **--delete-key** - удалить ключ;
- **--delete-secret-key** - удалить секретный ключ;
- **--export** - экспортовать все ключи;
- **--export-secret-keys** - экспортовать все секретные ключи;
- **--import** - импортировать ключи;
- **--send-keys** - отправить ключи на сервер, должен быть указан сервер ключей;
- **--recv-keys** - получить ключи от сервера ключей;
- **--keyserver** - указать сервер ключей;
- **--fetch-keys** - скачать ключи;
- **--gen-key** - создать ключ;
- **--sign-key** - подписать ключ;
- **--passwd** - изменить пароль для ключа.

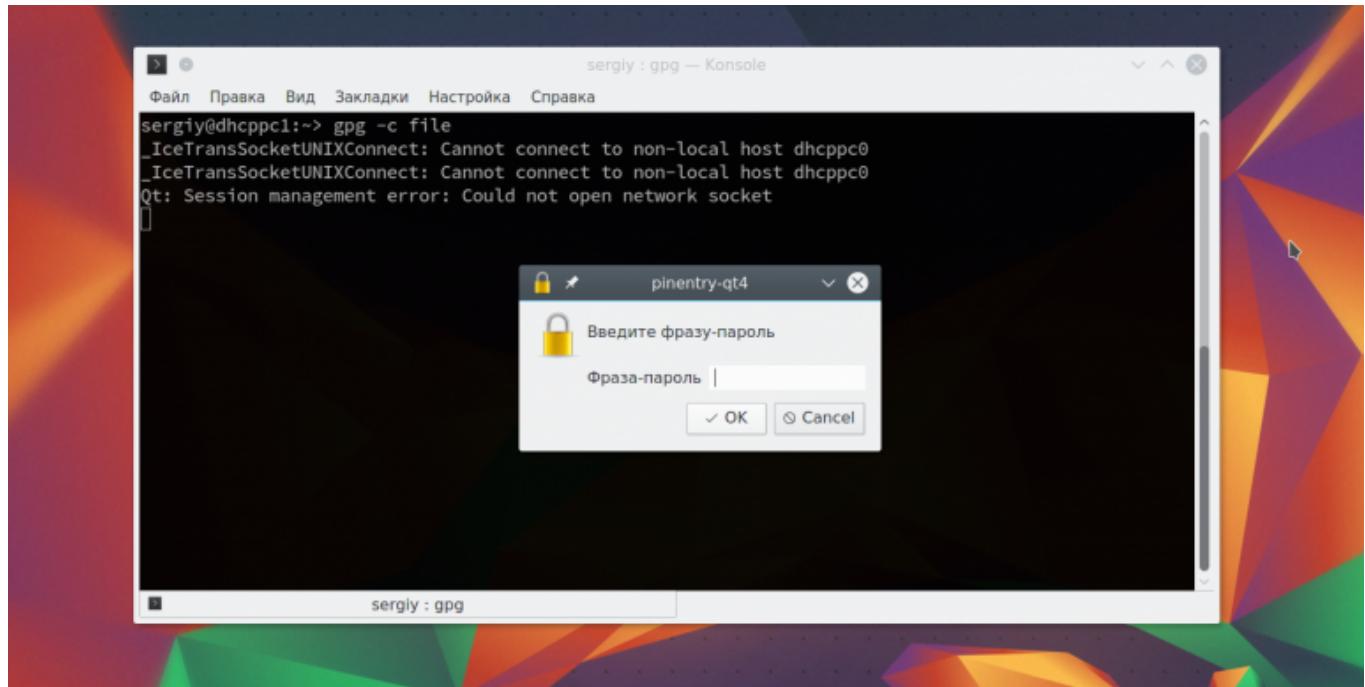
А теперь рассмотрим по порядку, что нам нужно для того, чтобы выполнять шифрование файлов Linux.

## Шифрование файлов с помощью пароля

[Privacy](#)

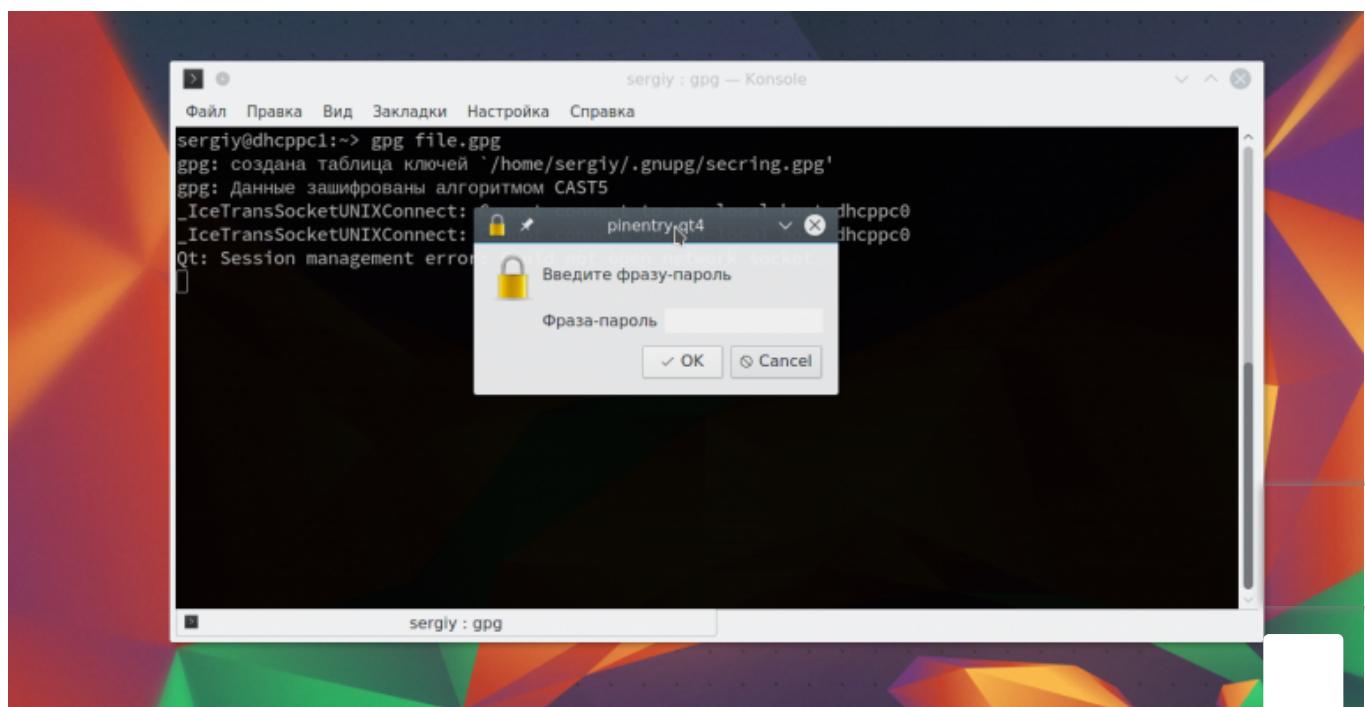
Симметричный шифр - самый простой и в то же время надежный способ шифрования файлов linux. Расшифровать файл сможет любой у кого есть пароль. Для использования просто запустите терминал и выполните команду gpg с параметром -c:

```
$ gpg -c имя файла
```



Утилита создаст файл с расширением gpg. Для расшифровки используйте:

```
$ gpg имя_файла.gpg
```



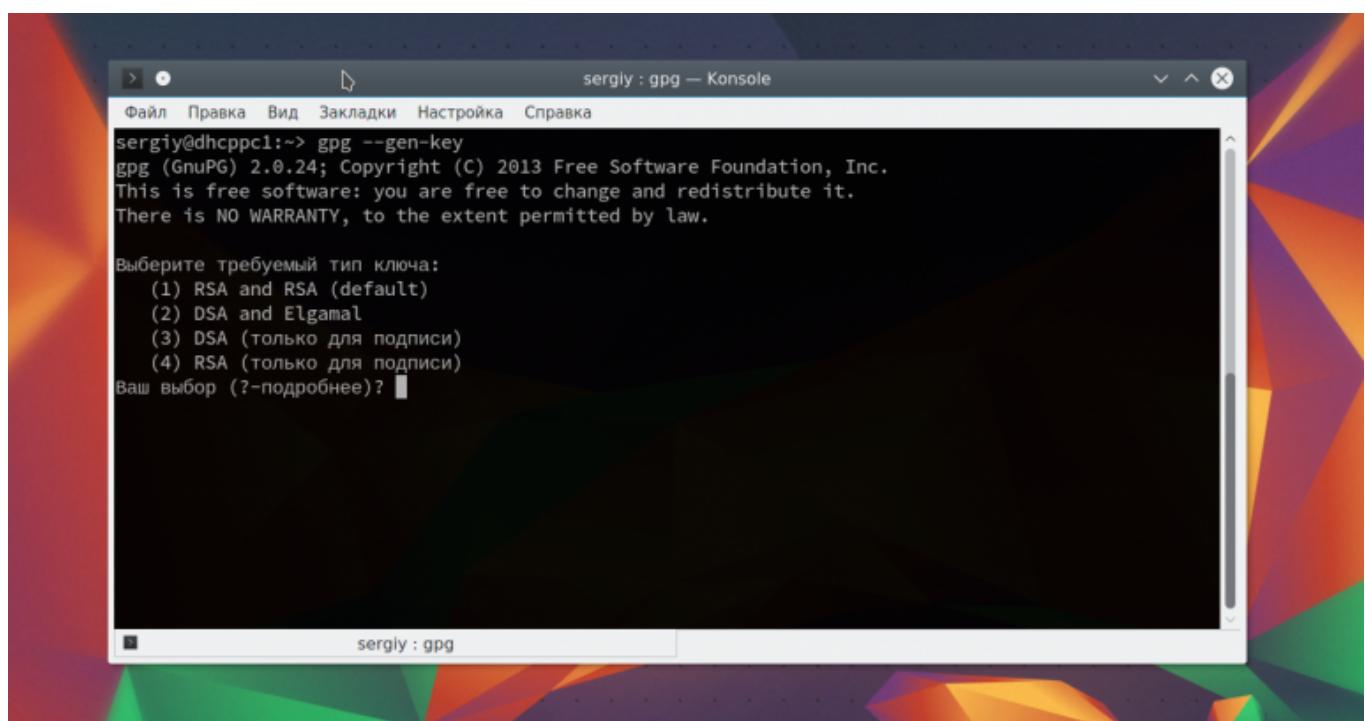
# Шифрование с использованием ключей

Асимметричный шифр более надежный так как для шифрования используется два ключа - публичный, собственно для шифрования, которым может воспользоваться любой, и приватный - для расшифровки. Причем файл можно расшифровать только с помощью приватного ключа, даже если вы зашифровали файл, без приватного ключа вы его не расшифруете.

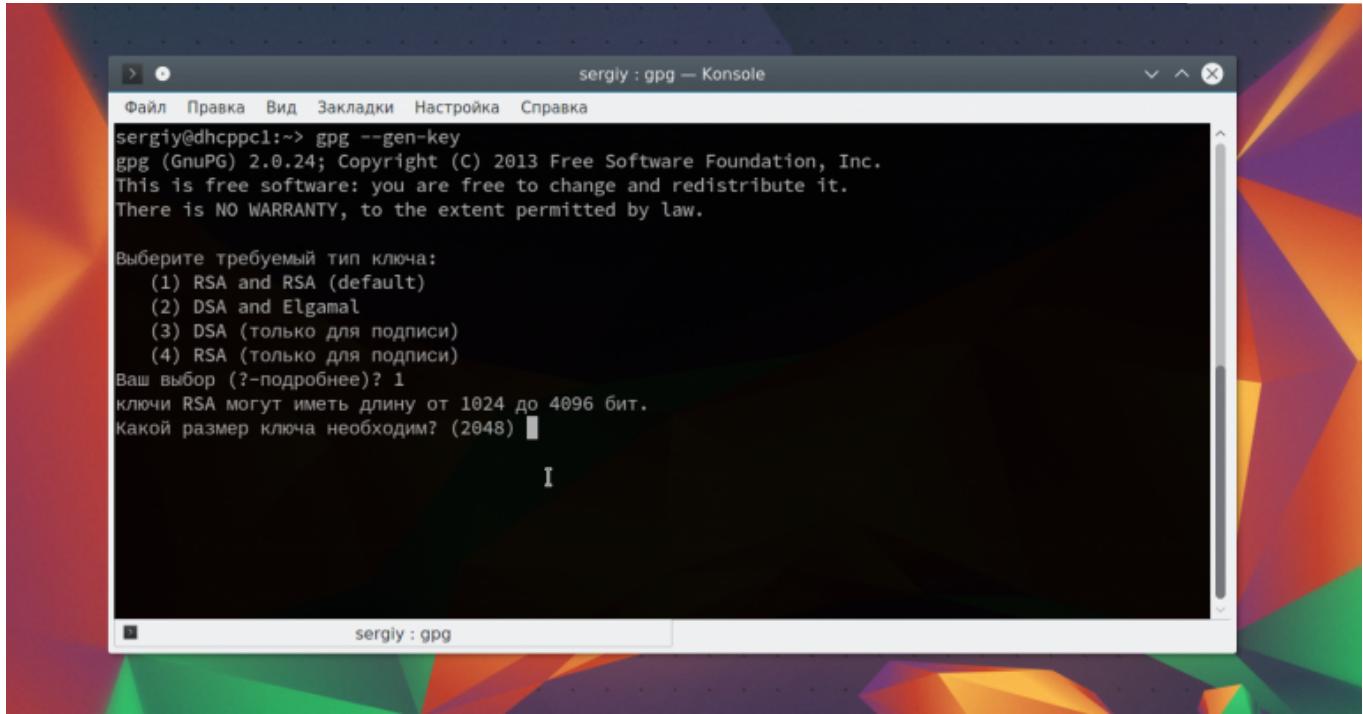
Сначала необходимо настроить gpg, создать пару ключей, для этого наберите:

```
$ gpg --gen-key
```

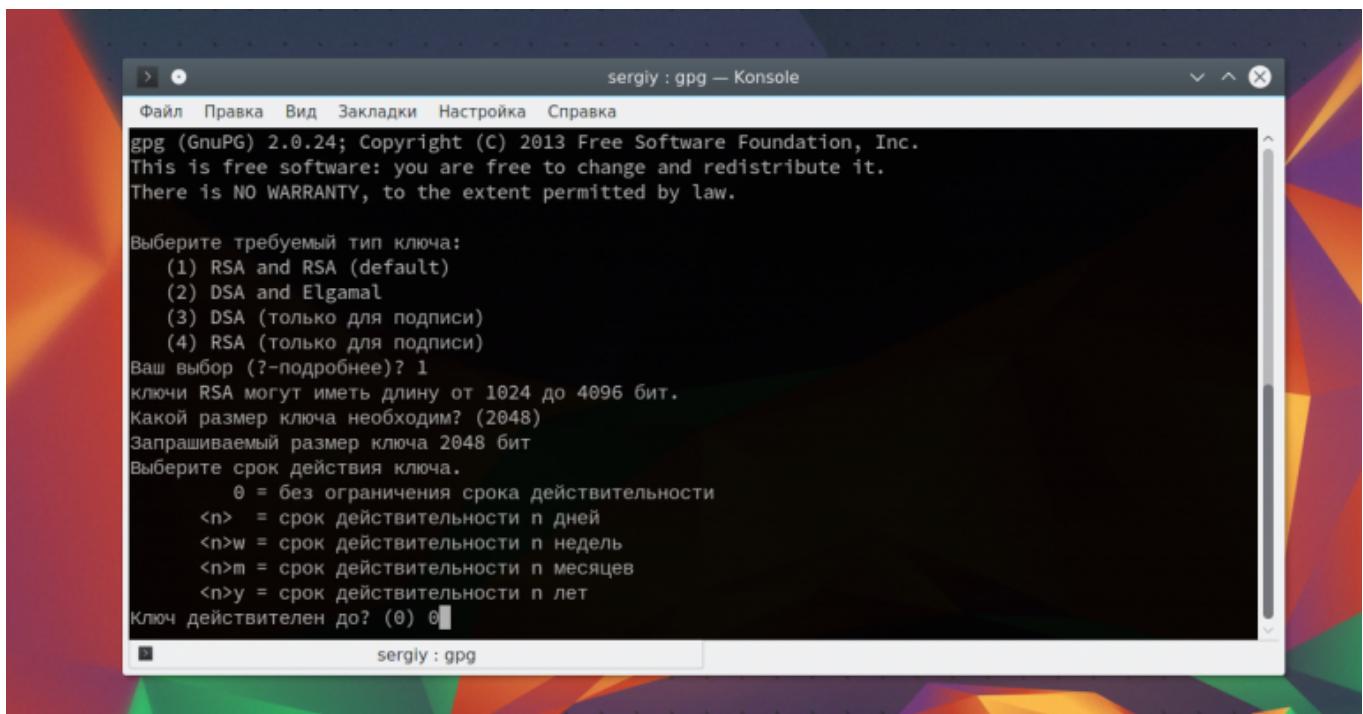
Программа задаст ряд вопросов для настройки ключа:



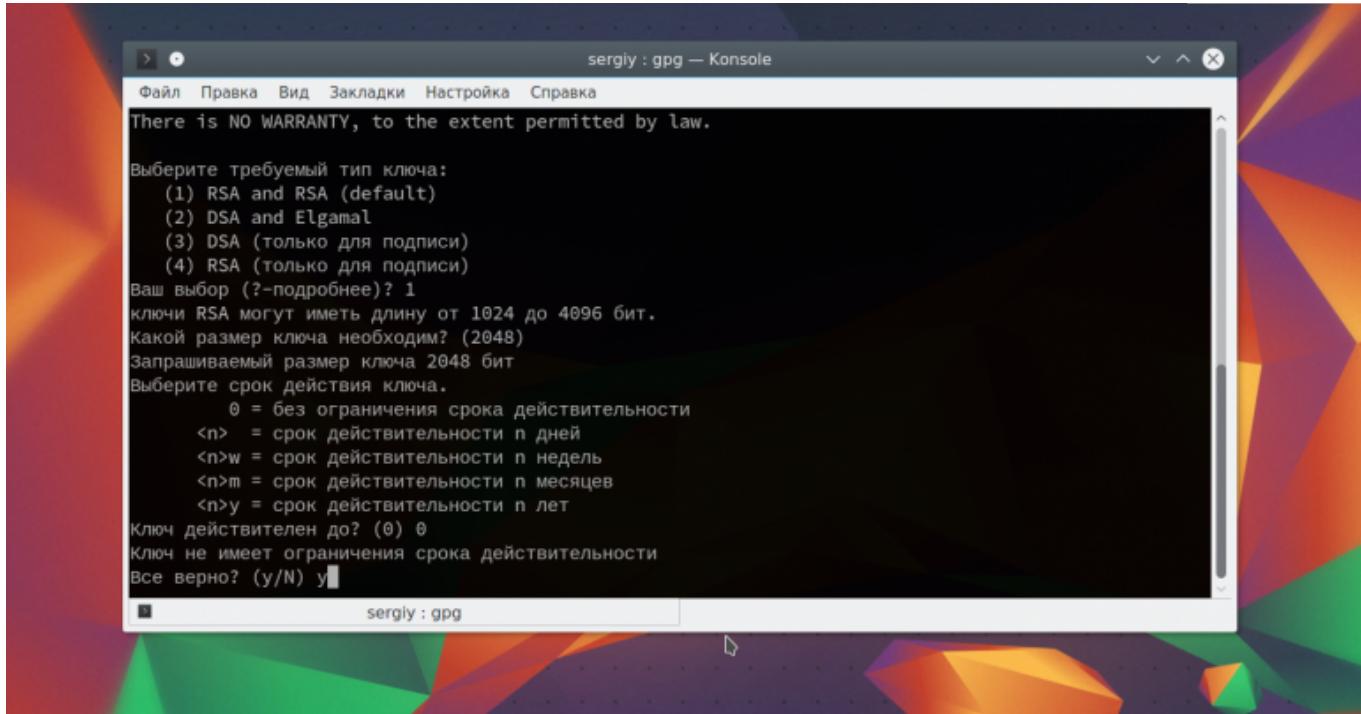
Выберите требуемый тип ключа.



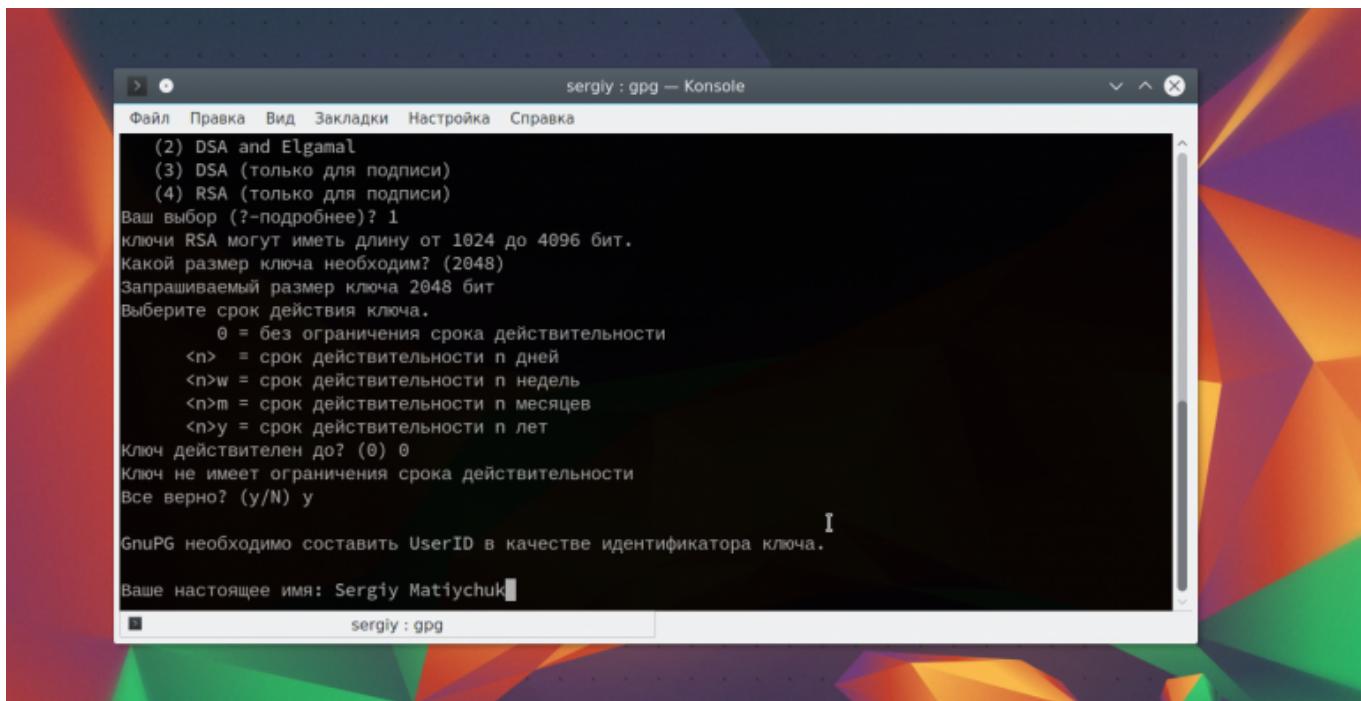
Выберите нужный размер для ключа, обычно 2048 будет достаточно.



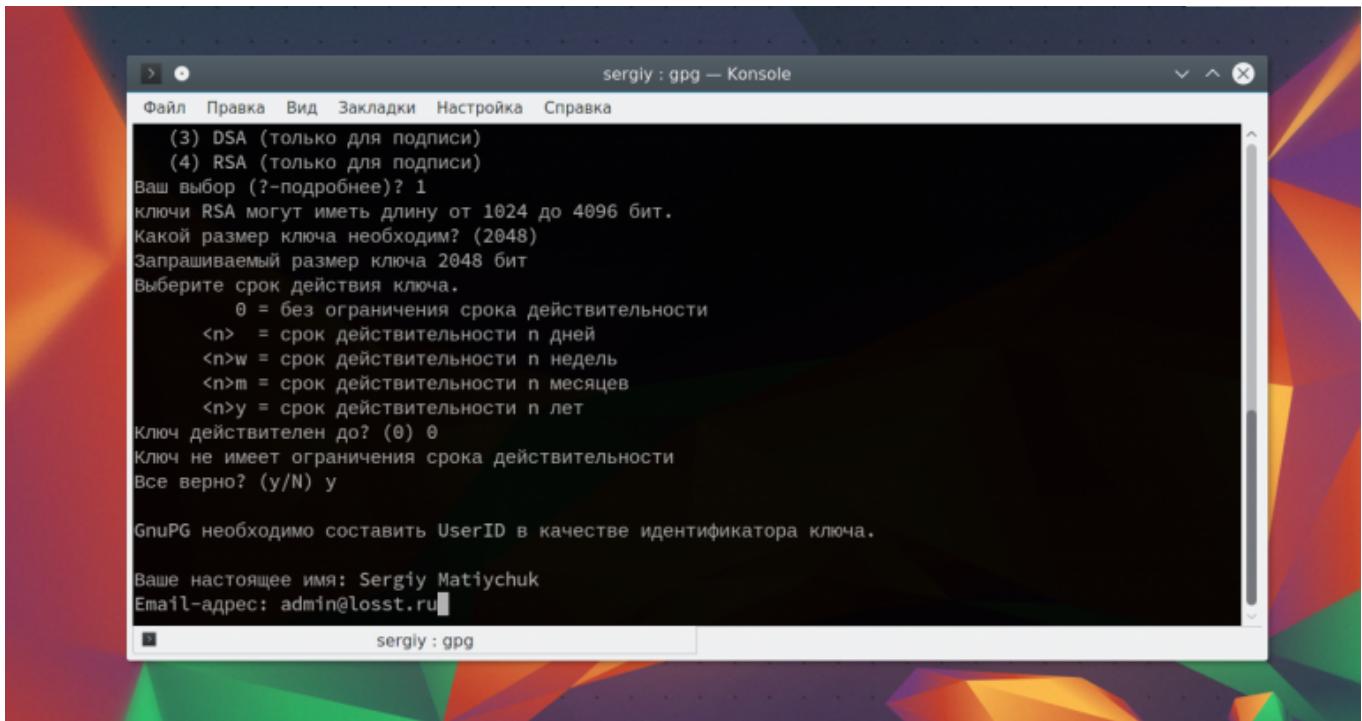
Выберите срок действия для ключа.



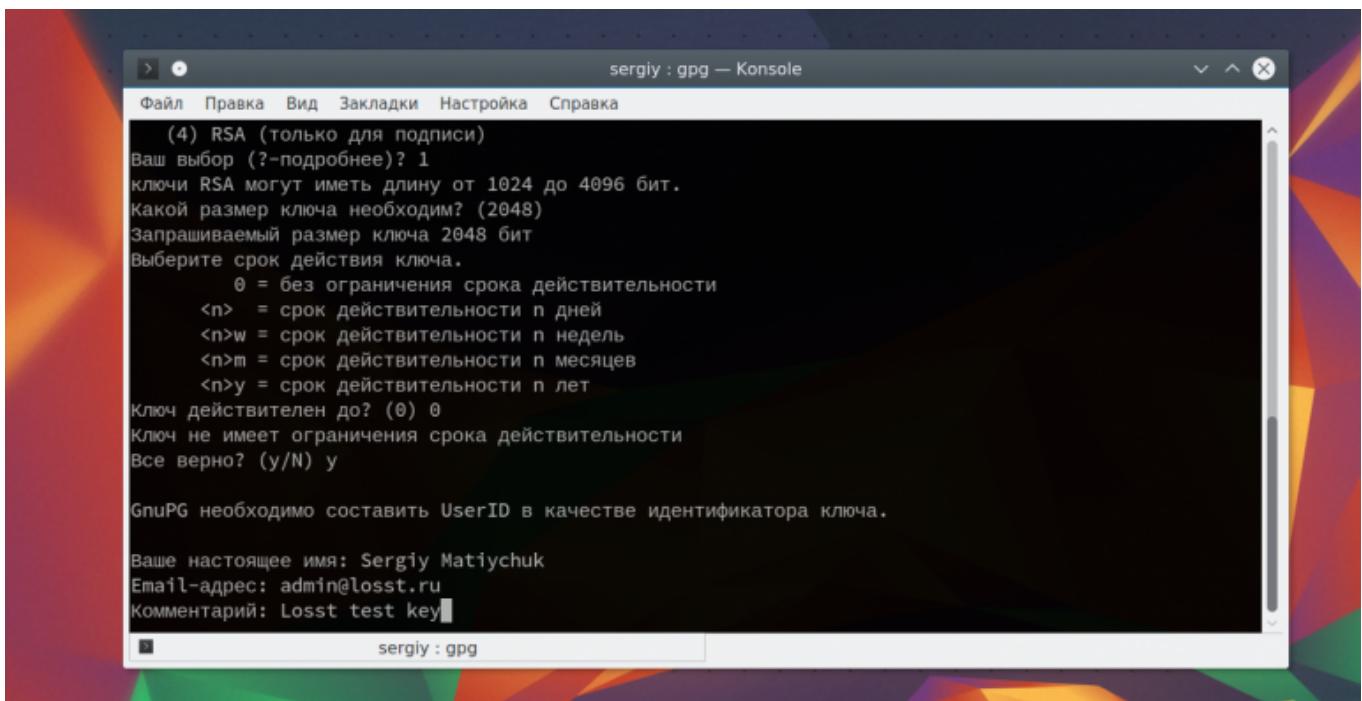
Проверьте все ли правильно.

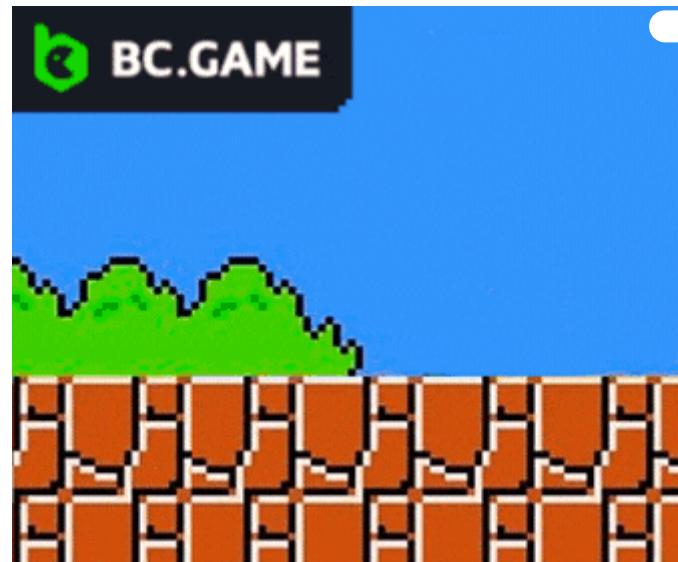


Введите имя нового ключа, фактически, это имя пользователя, но вы будете использовать его чтобы зашифровать файл linux, поэтому выбирайте обдумано.

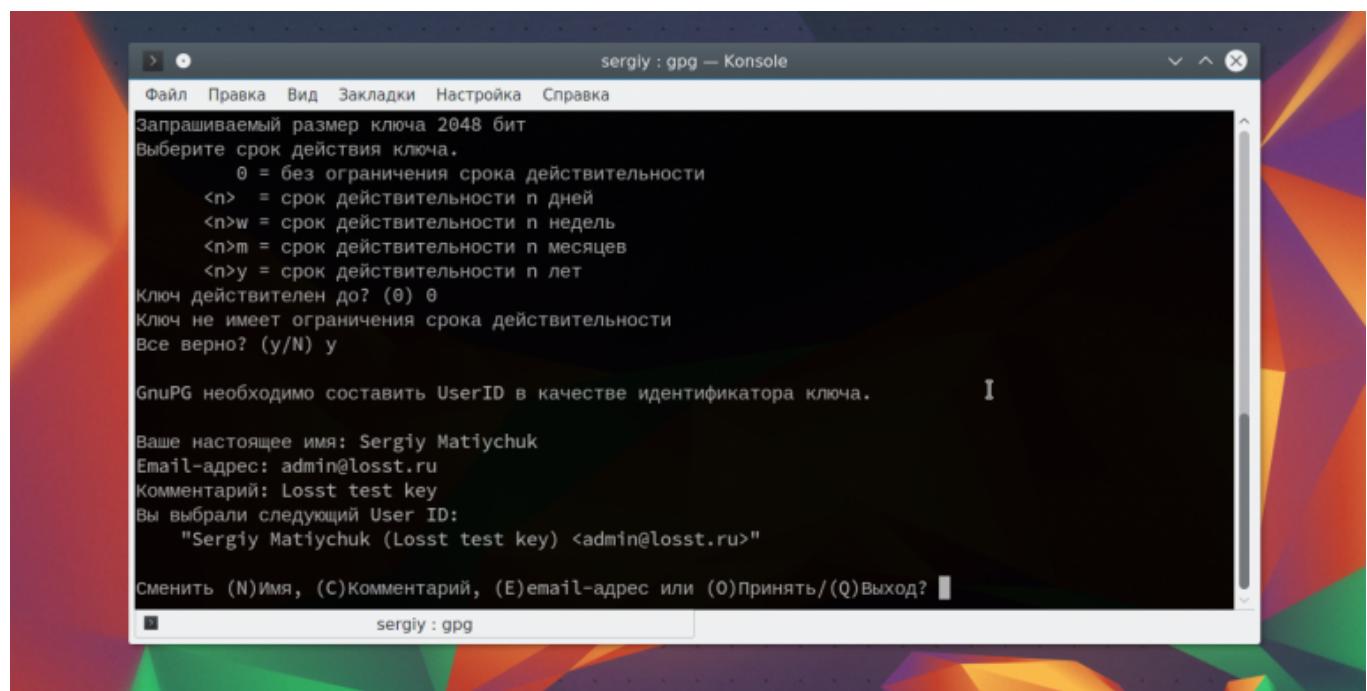


Введите ваш email адрес.

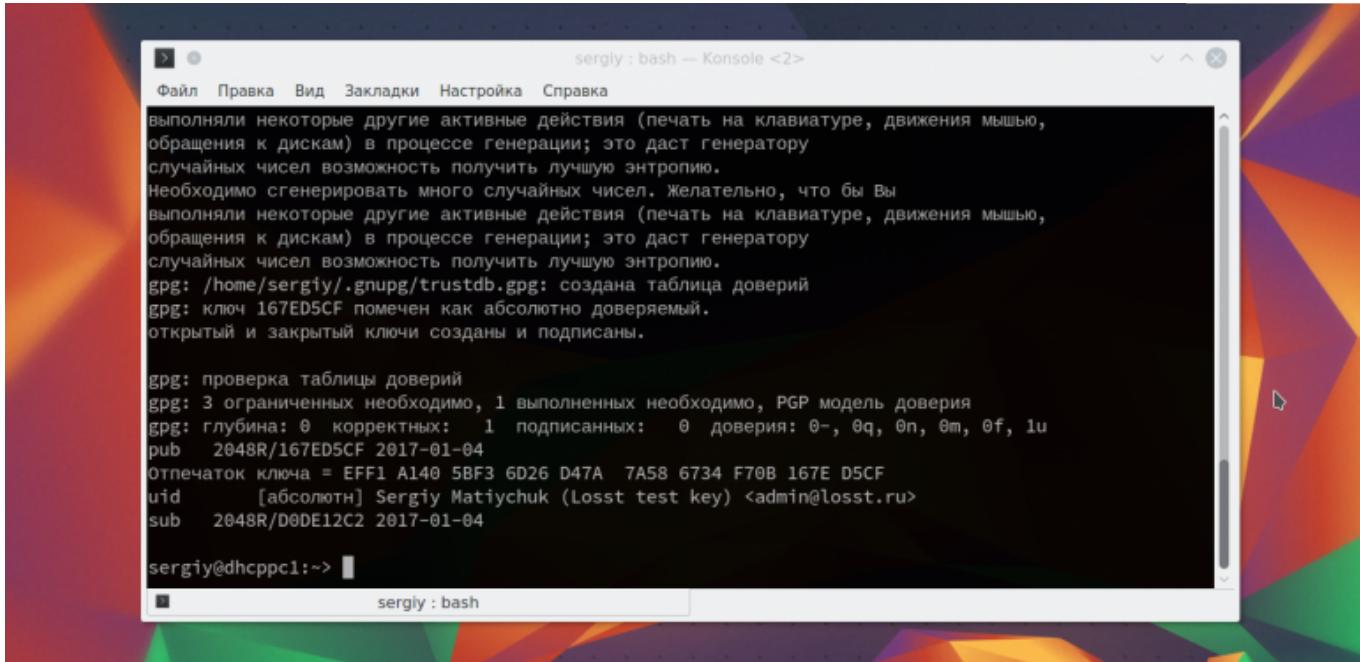




Описание ключа, если нужно.



Финальная проверка, затем нажмите 0 для завершения.



The screenshot shows a terminal window titled "sergiy : bash — Konsole <2>". The output of the gpg command is displayed:

```

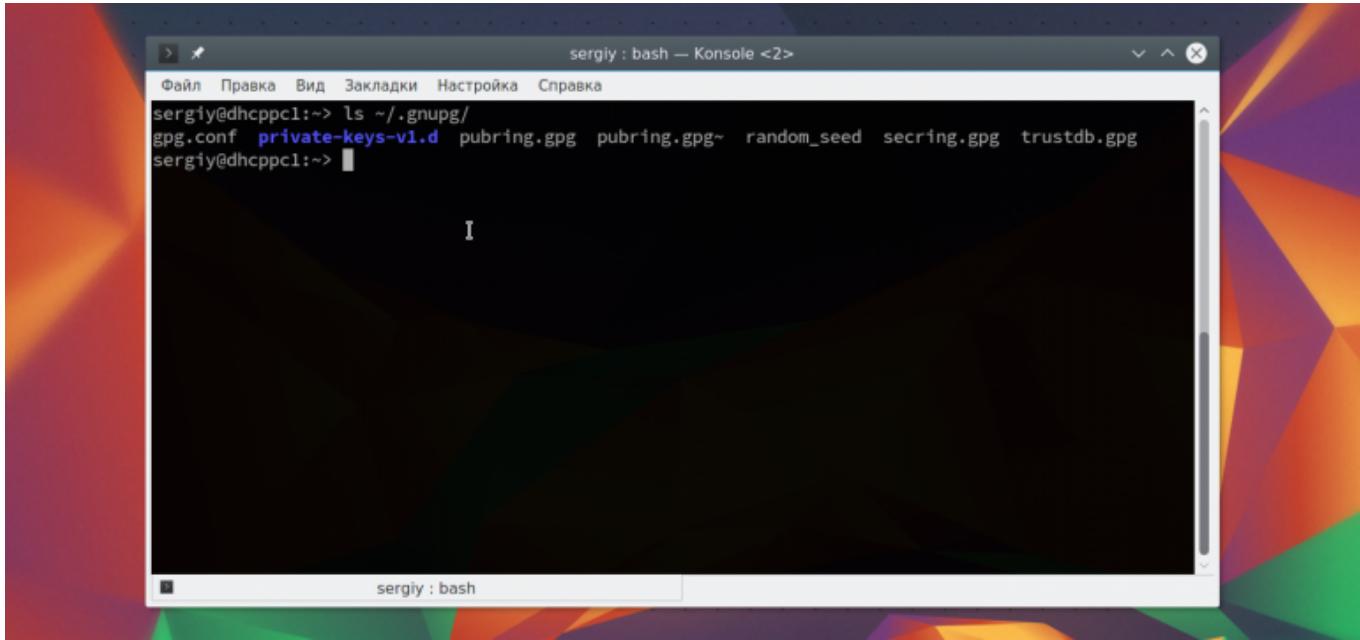
выполняли некоторые другие активные действия (печать на клавиатуре, движения мышью,
обращения к дискам) в процессе генерации; это даст генератору
случайных чисел возможность получить лучшую энтропию.
Необходимо сгенерировать много случайных чисел. Желательно, что бы Вы
выполняли некоторые другие активные действия (печать на клавиатуре, движения мышью,
обращения к дискам) в процессе генерации; это даст генератору
случайных чисел возможность получить лучшую энтропию.
gpg: /home/sergiy/.gnupg/trustdb.gpg: создана таблица доверий
gpg: ключ 167ED5CF 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: 1 подписанных: 0 доверия: 0-, 0q, 0n, 0m, 0f, 1u
pub 2048R/167ED5CF 2017-01-04
отпечаток ключа = EFF1 A140 5BF3 6D26 D47A 7A58 6734 F70B 167E D5CF
uid [абсолютн] Sergiy Matiychuk (Losst test key) <admin@losst.ru>
sub 2048R/D0DE12C2 2017-01-04

```

Процесс генерации может занять некоторое время. Когда все будет готово в каталоге `~/.gnupg` появятся два файла. В файле `pubring.gpg` публичный ключ, а в `secring.gpg` приватный.

`$ ls ~/.gnupg/`



The screenshot shows a terminal window titled "sergiy : bash — Konsole <2>". The output of the `ls` command in the `~/.gnupg` directory is displayed:

```

sergiy@dhcppc1:~> ls ~/.gnupg/
gpg.conf private-keys-v1.d pubring.gpg pubring.gpg~ random_seed secring.gpg trustdb.gpg
sergiy@dhcppc1:~>
```

Также вы можете посмотреть список доступных ключей:

`$ gpg --list-keys`

```
sergiy@dhcppc1:~> gpg --list-keys
/home/sergiy/.gnupg/pubring.gpg
-----
pub 2048R/167ED5CF 2017-01-04
uid [абсолютн] Sergiy Matiychuk (Losst test key) <admin@losst.ru>
sub 2048R/D0DE12C2 2017-01-04
sergiy@dhcppc1:~>
```

Если вы собираетесь шифровать файлы на другом компьютере необходимо экспортировать публичный ключ, для этого есть опция -a:

```
$ gpg -a -o gpgkey.asc --export имя_ключа
```

```
sergiy@dhcppc1:~> gpg --list-keys
/home/sergiy/.gnupg/pubring.gpg
-----
pub 2048R/167ED5CF 2017-01-04
uid [абсолютн] Sergiy Matiychuk (Losst test key) <admin@losst.ru>
sub 2048R/D0DE12C2 2017-01-04

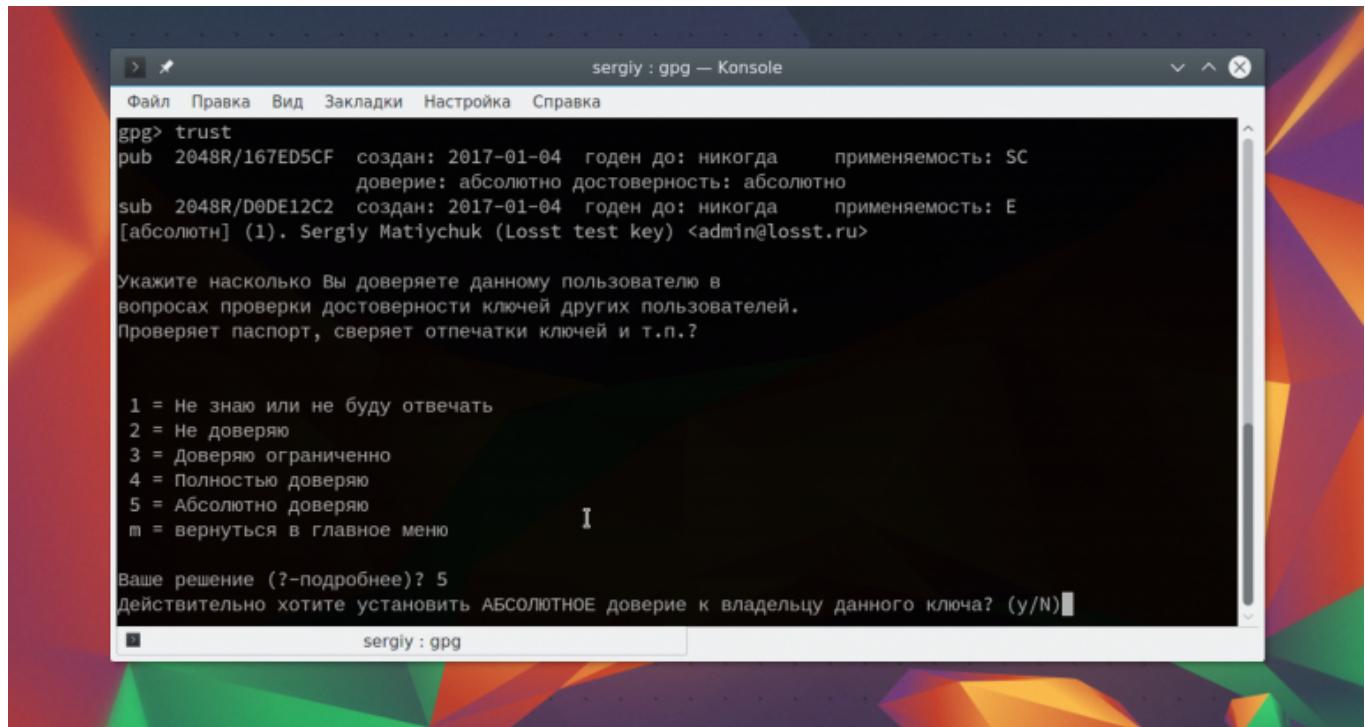
sergiy@dhcppc1:~> gpg -a -o gpgkey.asc --export Sergiy
sergiy@dhcppc1:~> ls gpgkey.asc
gpgkey.asc
sergiy@dhcppc1:~>
```

Затем передаем файл на целевое устройство и импортируем ключ:

```
$ gpg --import gpgkey.asc
```

После импорта ключа уровень доверия к нему по умолчанию будет неизвестным поэтому при каждом шифровании gpg будет спрашивать действительно ли вы доверяете этому ключу. Чтобы этого избежать нужно указать уровень доверия. Для этого воспользуйтесь редактором ключей:

```
$ gpg --edit-key Username
```



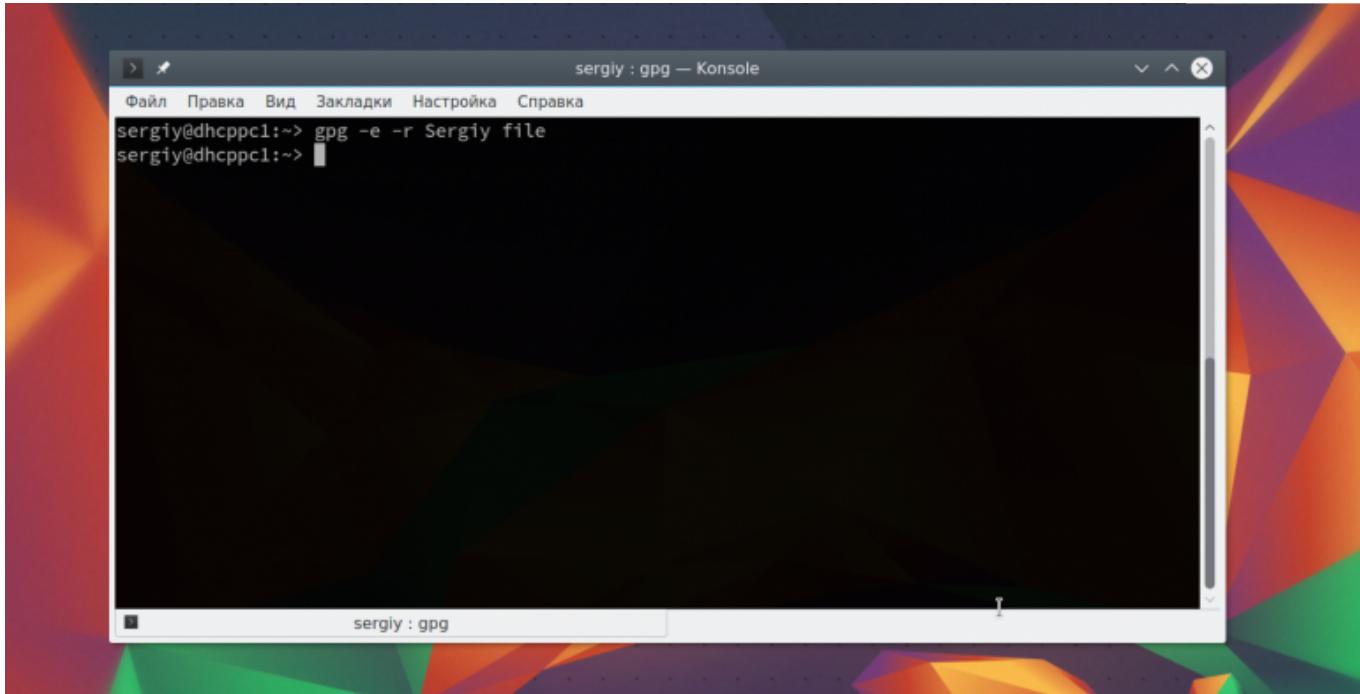
Для выбора уровня доверия введите команду `trust`:

```
gpg> trust
```

Для своих ключей можно использовать пункт абсолютно доверяю с номером 5, вы же знаете что это именно ваш ключ.

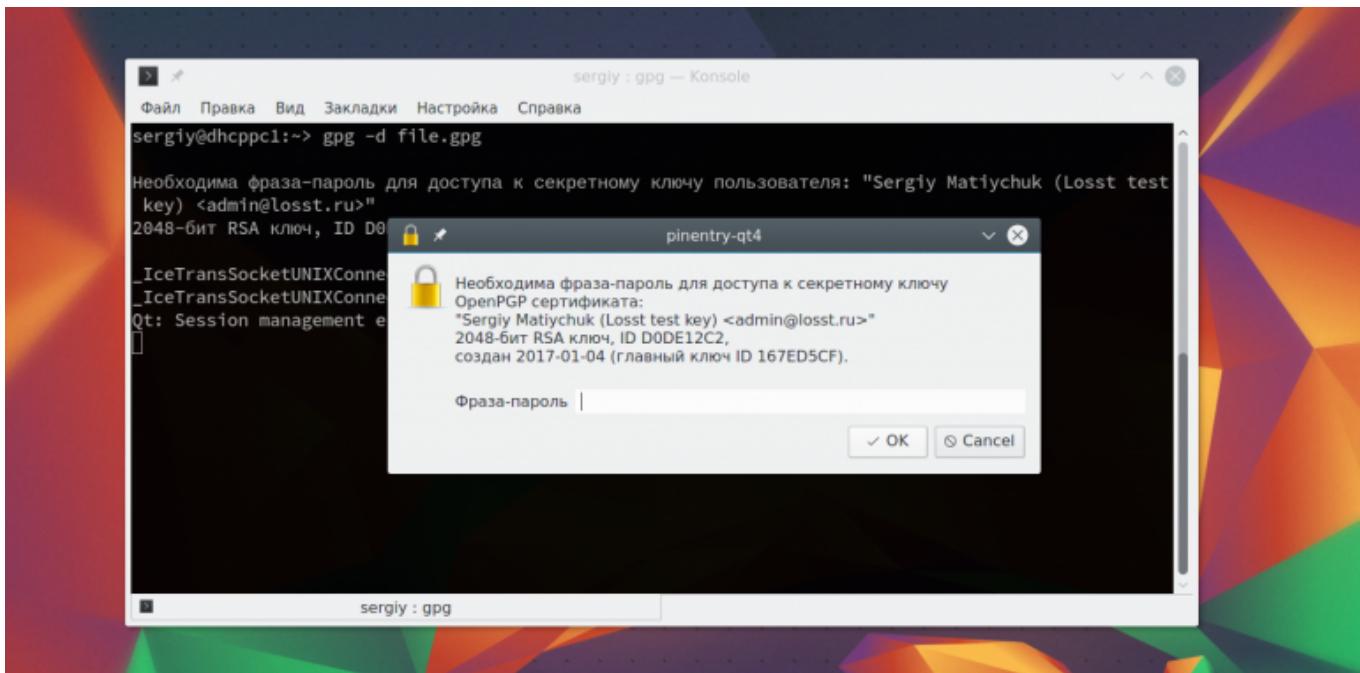
Теперь можно переходить к шифрованию. Для того чтобы зашифровать файл `linux` используйте команду:

```
$ gpg -e -g ид_пользователя имя_файла
```



Ид пользователя нужно указывать тот что вы использовали при создании ключа. Для расшифровки используйте:

```
$ gpg -d имя_файла.gpg
```



Для каталогов действия аналогичны только сначала нужно создать архив с помощью tar:

```
tar -cf - каталог | gpg -e -r ИД_пользователя
```

А для расшифровки:

Privacy

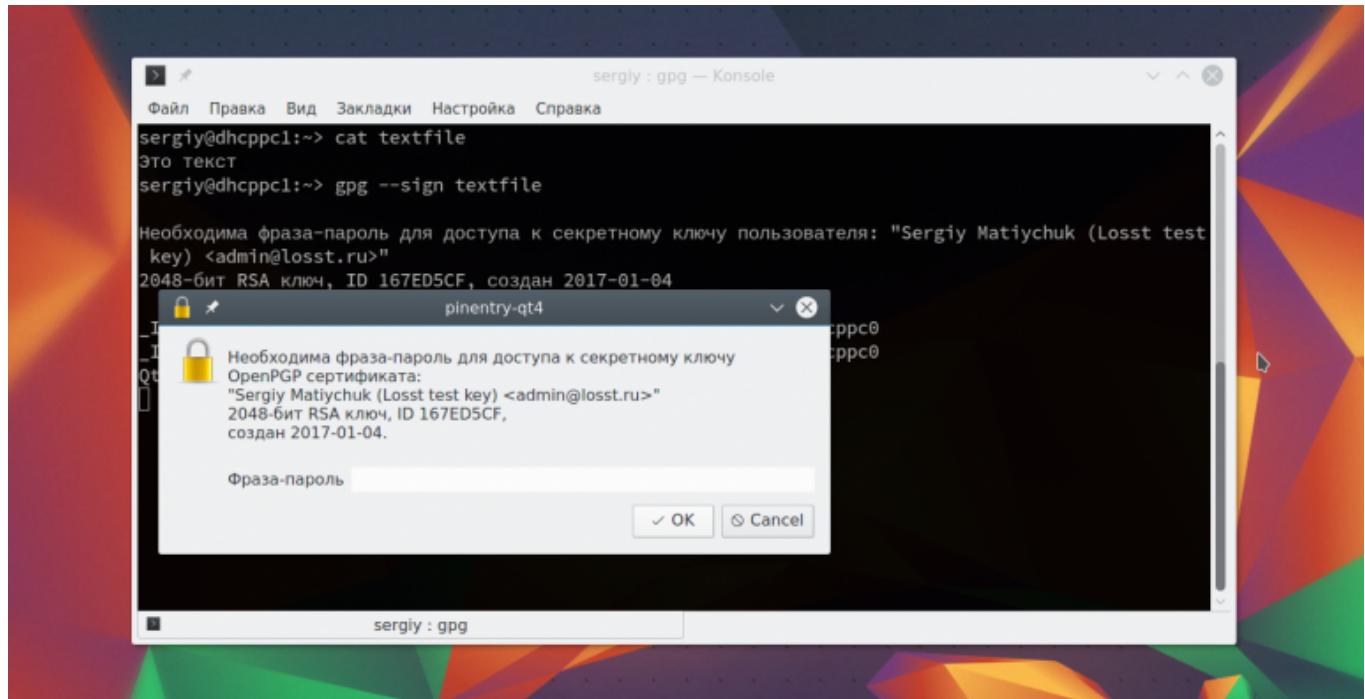
```
$ gpg -d каталог.gpg | tar -xvf
```

## Подписи и шифрование

Для проверки подлинности файлов может использоваться не шифрование, а подпись. Тогда на основе файла и ключа создается отпечаток, который записывается в файл. Если файл будет изменен, то отпечаток уже не совпадет.

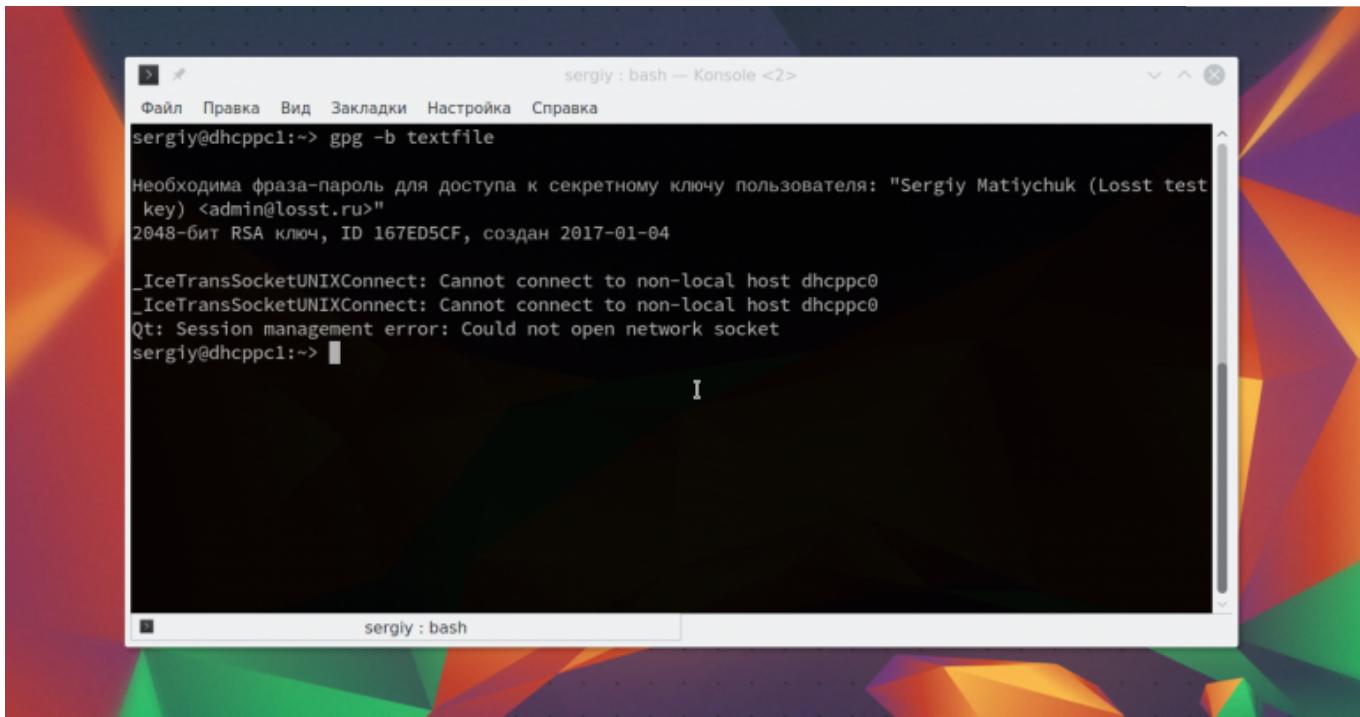
Вы можете подписать файл с помощью опции `--sign`:

```
$ gpg --sign имя_файла
```



Если вы не хотите изменять исходный файл, то можно создать подпись в отдельном файле:

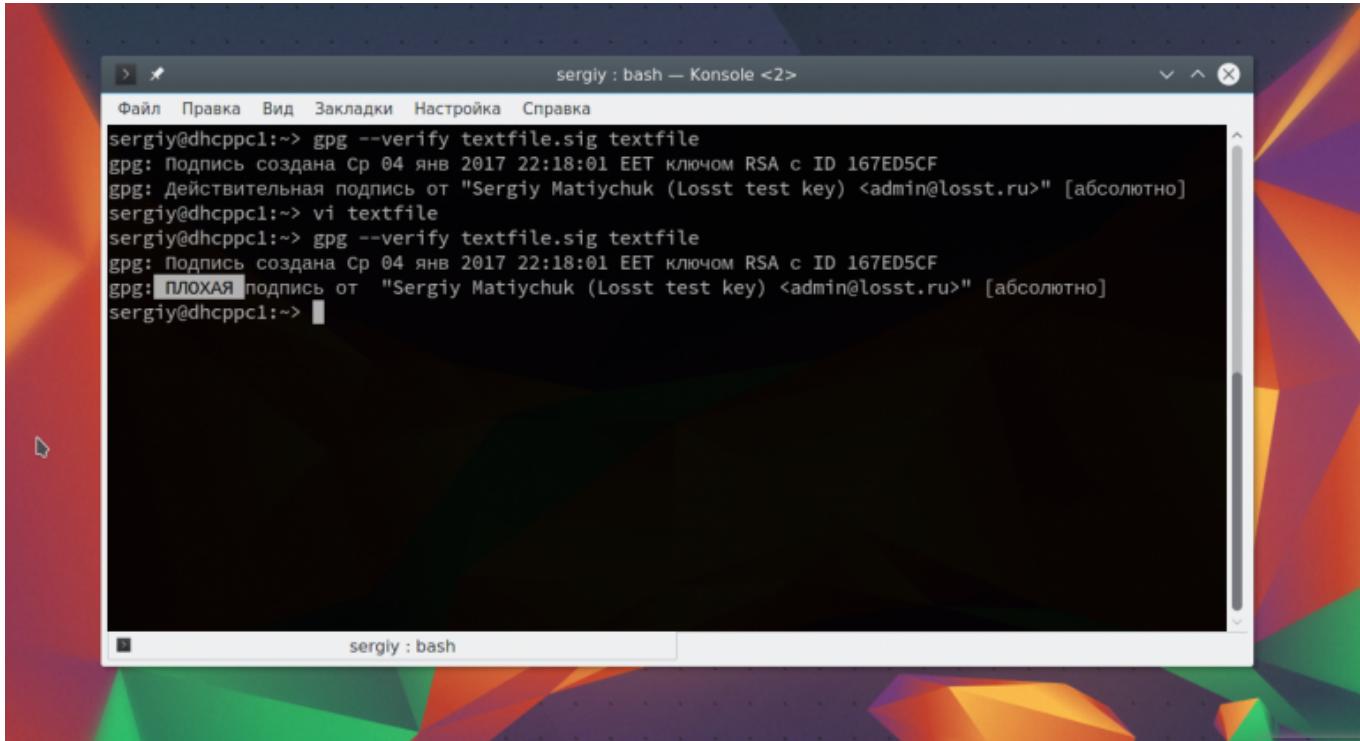
```
$ gpg -b имя_файла
```



```
sergiy : bash — Konsole <2>
Файл Правка Вид Закладки Настройка Справка
sergiy@dhcpc1:~> gpg --gen-key
Необходима фраза-пароль для доступа к секретному ключу пользователя: "Sergiy Matiychuk (Losst test key) <admin@losst.ru>"  
2048-бит RSA ключ, ID 167ED5CF, создан 2017-01-04
_IceTransSocketUNIXConnect: Cannot connect to non-local host dhcpc0
_IceTransSocketUNIXConnect: Cannot connect to non-local host dhcpc0
Qt: Session management error: Could not open network socket
sergiy@dhcpc1:~>
```

Тогда в каталоге, рядом с файлом появится файл .sig с подписью. Дальше, чтобы проверить достаточно использовать команду verify:

```
$ gpg --verify textfile.sig textfile
```



```
sergiy : bash — Konsole <2>
Файл Правка Вид Закладки Настройка Справка
sergiy@dhcpc1:~> gpg --verify textfile.sig textfile
gpg: Подпись создана Ср 04 янв 2017 22:18:01 EET ключом RSA с ID 167ED5CF
gpg: Действительная подпись от "Sergiy Matiychuk (Losst test key) <admin@losst.ru>" [абсолютно]
sergiy@dhcpc1:~> vi textfile
sergiy@dhcpc1:~> gpg --verify textfile.sig textfile
gpg: Подпись создана Ср 04 янв 2017 22:18:01 EET ключом RSA с ID 167ED5CF
gpg: ПЛОХАЯ подпись от "Sergiy Matiychuk (Losst test key) <admin@losst.ru>" [абсолютно]
sergiy@dhcpc1:~>
```

Если файл был изменен, то вы увидите, что подпись не сходиться.

## Выводы

Privacy

В этой статье мы рассмотрели как выполняется шифрование файла `linux`, а также настройка утилиты `gpg`. Шифрование `gpg` `linux` используется людьми для хранения важных данных, а механизм подписей популярен среди разработчиков дистрибутивов. Если у вас остались вопросы, спрашивайте в комментариях!

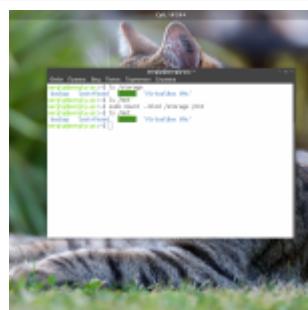
На завершение отличное видео о том, как работает асимметричный алгоритм шифрования:

Была ли эта информация полезной для вас?  Да  Нет X

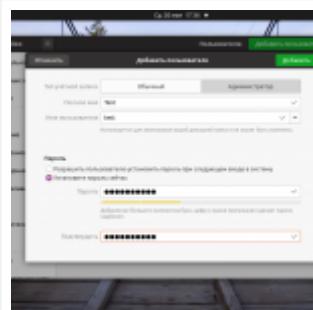
## Похожие записи



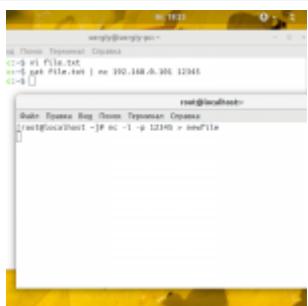
Шифрование файлов  
для облака в



Монтирование папок  
в Linux



Шифрование  
домашней папки в



Передача файлов по  
сети Linux

## Оцените статью

(5 оценок, среднее: 5,00 из 5)

[Инструкции](#)

[ubuntu](#), [шифрование](#)

## Об авторе



ADMIN

Основатель сайта losst.pro, увлекаюсь открытым программным обеспечением и операционной системой Linux. В качестве основной ОС сейчас использую Ubuntu. Кроме Linux, интересуюсь всем, что связано с информационными технологиями и современной наукой.

[Privacy](#)

## 5 комментариев к “Шифрование файлов и папок в Linux”



Turion

2 сентября, 2016 в 7:19 дп

Как расшифровывать файлы на другом компьютере (windows)

[Ответить](#)



Ignat

15 января, 2019 в 12:03пп

Название не соответствует заявленному контенту. Удалить.

[Ответить](#)



Хрущ

10 мая, 2022 в 3:47пп

некера не шифруется. почему нет описание что такое команда -г? или тут одни гиги вас читаю по умолчанию. при наборе gpg -e -г ид\_пользователя имя\_файла. много раз послан нахер. шел 2022 год линукс побеждал винду где то в паралельной вселеной

[Ответить](#)



dimame185

12 января, 2024 в 11:40пп

--recipient имя

-Г

Шифровать для идентификатора пользователя имя. Если не указан ни этот

Privacy

параметр, ни --hidden-recipient, GnuPG запрашивает идентификатор пользователя, если не задан --default-recipient.

[Ответить](#)



Андрей

5 июля, 2022 в 12:04 pp

Дополните пж, что для расшифровки файла нужно использовать вот так:

gpg --d имя\_зашифрованного\_файла > расшифрованный\_файл

Иначе оно вываливает в консоль, это не очень. Особенно, если расшифровывать архив.

[Ответить](#)

## Оставьте комментарий

Имя \*

Email

Я прочитал и принимаю политику конфиденциальности. Подробнее [Политика конфиденциальности](#) \*

Комментировать

Privacy

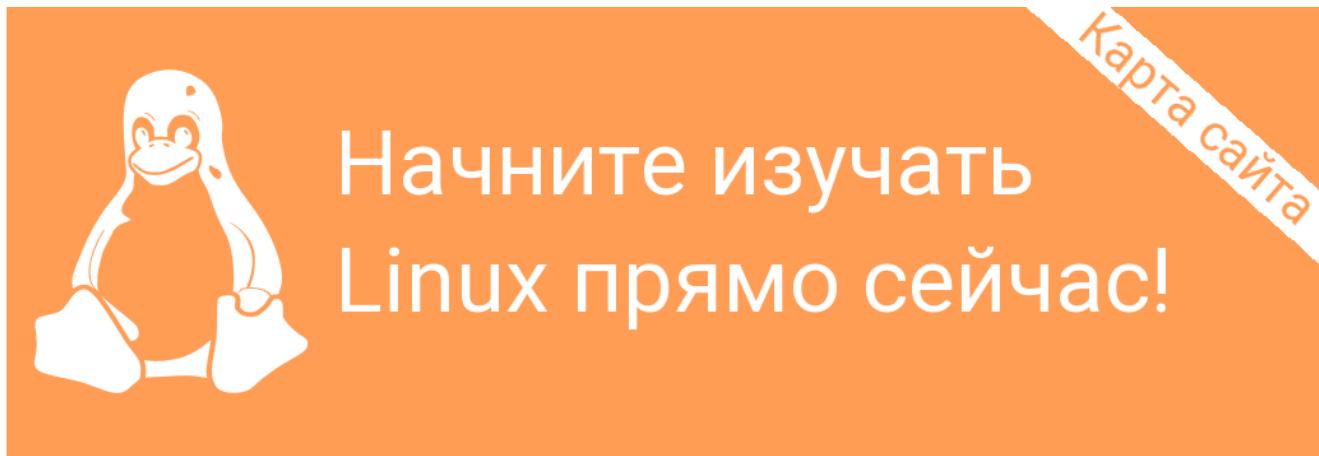
Русский

Поиск

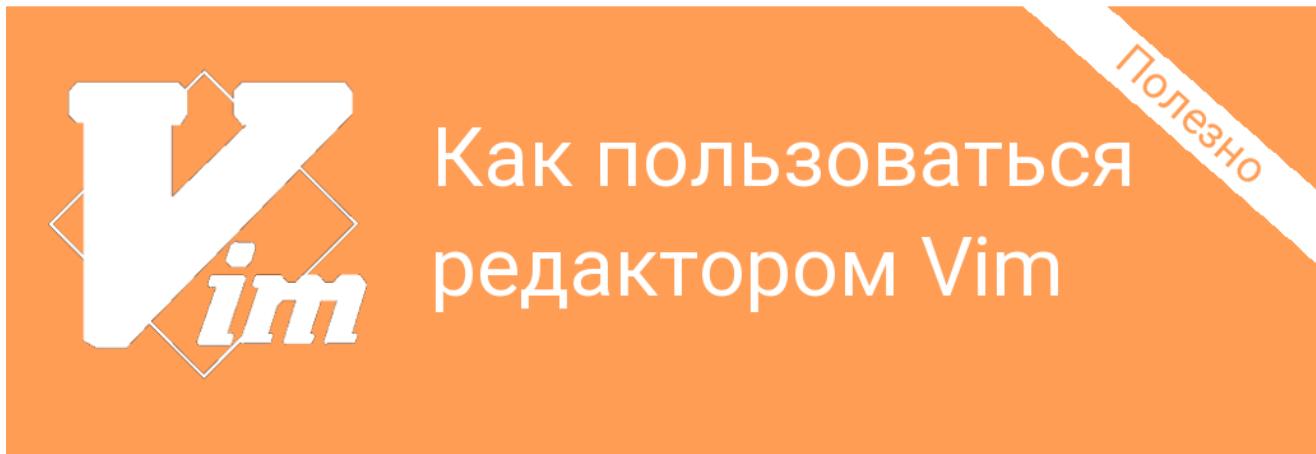
## ПОИСК ПО КОМАНДАМ

Начните вводить команду

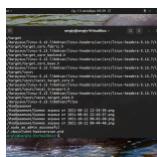
Поиск



Privacy

[Лучшие](#)[Свежие](#)[Теги](#)[Команда chmod в Linux](#)

2020-04-13

[Команда find в Linux](#)

2021-10-17

[Как узнать IP-адрес в Linux](#)

2023-04-14

[Настройка Старт](#)

2021-10-01

[Права доступа к файлам в Linux](#)

2020-10-09



Privacy

# РАССЫЛКА

Ваш E-Mail адрес

Я прочитал(а) и принимаю политику конфиденциальности

Sign up



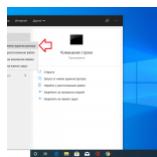
Windows

Списки



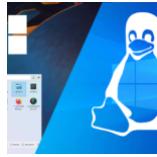
Подключение Ext4 в Windows

2023-02-18



Восстановление Grub после установки Windows 10

2020-08-15

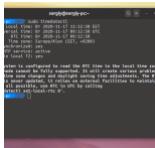


Установка Linux рядом с Windows 10 или 11

2023-02-08

Сбиваются время в Ubuntu и Windows

Privacy



2023-02-18

[Смотреть ещё](#)

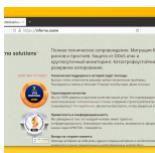
## МЕТА

[Регистрация](#)
[Войти](#)
[Лента записей](#)
[Лента комментариев](#)

## СЛЕДИТЕ ЗА НАМИ В СОЦИАЛЬНЫХ СЕТЯХ

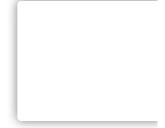


## Интересное



Лучшие VPS сервера для VPN

2022-09-04

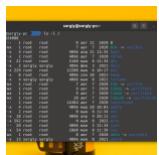


Privacy



## Команды терминала Linux

2020-12-18



## Как правильно: папка или каталог в Linux

2022-05-10



## 6 причин, почему Ubuntu лучше Windows

2021-10-01

©Losst 2024 CC-BY-SA [Политика конфиденциальности](#)

Privacy