Шифрование сообщений с помощью средств GNU Privacy Guard

Инсталлируйте вторую версию GnuPG на ваш компьютер:
sudo apt-get install gnupg2

Выведите на консоль информацию о текущей версии GnuPG и поддерживаемых криптоалгоритмах gpg2 --version
Набор основных команд можно увидеть по gpg2 –help

Приступайте к генерации пары (private и public) ключей:
gpg2 --gen-key
Выберите тип ключа 1 , чтобы была возможность и шифровать сообщения и подписывать их
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Можно выбрать 4096, это будет надежнее, но генерация продлится дольше.
Please specify how long the key should be valid.
Выберите время жизни ключа, например, несколько недель 3w.
GnuPG needs to construct a user ID to identify your key.
Введите ваш идентификатор, потом GnuPG еще попросит ввести ваш email и Comment. Это необходимо для идентификации ваших ключей.
You need a Passphrase to protect your secret key.
Введите пароль от закрытого (private) ключа и ЗАПОМНИТЕ его.
Теперь запустится процесс генерации ключевой пары, который будет длиться некоторое время, причем GnuPG будет вываливать просьбы:
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.
которые можно выполнять, что, в принципе, может ускорить процесс генерации ключей.

Если,вдруг,появится сообщение
Not enough random bytes available. Please do some other work to give the OS a chance to collect more entropy! (Need 204 more bytes)
Тогда вам придется инсталлировать еще и демона для сбора энтропии:
sudo apt-get install rng-tools

По завершении процесса генерации появится информация подобная этой
gpg: key 8640D6B9 marked as ultimately trusted
public and secret key created and signed.
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0 valid: 1 signed: 0 trust: 0-, 0q, 0n, 0m, 0f, 1u
gpg: next trustdb check due at 2020-03-24
pub 4096R/8640D6B9 2020-02-25 [expires: 2020-03-24]

Key fingerprint = DB5E AA39 0745 427D ED31 D189 3197 3F00 8640 D6B9
uid Alexeev (March_24) <afiskon@example.com>
sub 4096R/5982B4BF 2020-02-25 [expires: 2020-03-24]

Получить fingerprint (отпечаток) ключа можно gpg2 –fingerprint <e-mail>
Нужны fingerprints для того, чтобы в дальнейшем, проверять, что с сервера ключей был импортирован, действительно, правильный ключ.
Просмотреть список ключей можно командой gpg2 –list-keys

Удалить ключи (если такое понадобится) можно, например :
gpg2 --delete-secret-keys 8640D6B9
gpg2 --delete-keys 8640D6B9

Зайдите в подкаталог .gnupg вашего Home каталога и проанализируйте его содержимое в результате генерации ключей.
Создайте в вашем Home каталоге рабочий подкаталог для упражнений с шифрованием файлов и подписями. Зайдите в него и создайте текстовый файл, например, original.txt с каким-либо содержимым для шифрования.

Выполните шифрование на вашем ключе:
gpg2 -a -r 8640D6B9 -e original.txt
Просмотрите содержимое созданного в результате файла original.txt.asc, содержащего зашифрованную информацию cat original.txt.asc

Дешифрируйте текст с помощью
gpg2 -r 8640D6B9 -d original.txt.asc > decrypted.txt
You need a passphrase to unlock the secret key for
Введите пароль, закрытого (private) ключа (что был на этапе генерации ключей).

Просмотрите содержимое созданного в результате файла decrypted.txt
cat decrypted.txt и убедитесь в его совпадении с исходным файлом original.txt .

Создайте цифровую подпись текстового файла
gpg2 -r 8640D6B9 --clearsign message.txt
Цифровая подпись попадает в файл message.txt.asc содержимое просматривается командой cat message.txt.asc .
Проверка цифровой подписи выполняется, как
gpg2 --verify message.txt.asc

Самостоятельно исследуйте режимы работы GnuPG с различными опциями и ключами и выполнение команд из базового списка.