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.

Импорт и экспорт ключей
Цифровая подпись

Экспорт открытого ключа

Для того, чтобы нам могли шифровать файлы/сообщения, а также проверять наши подписи, мы должны экспортировать свой открытый ключ в файл, (либо на ключевой сервер).
Экспорт открытого ключа в текстовый файл:
gpg2 --export --armor 29FEB0EF > mykey.asc
Здесь **0x**29FEB0EF — отпечаток ключа нашей ключевой пары, открытый ключ которой экспортируем,
а mykey.asc — имя файла, в который будет сохранён результат.

Создание электронной цифровой подписи (ЭЦП) файла

GnuPG позволяет использовать несколько типов подписей:

- встроенная в файл:

содержимое файла изменяется так, чтобы в него была добавлена ЭЦП. Чаще всего применяется при отправке подписанных сообщений по электронной почте;

- отсоединённая в текстовом формате:

создаётся файл с расширением *.asc вида mydocument.pdf.asc (где mydocument.pdf — имя оригинального файла);

- отсоединённая в двоичном формате:

создаётся файл с расширением *.sig вида mydocument.pdf.sig в бинарном формате.
Для создания ЭЦП файла используется закрытый ключ из нашей ключевой пары, а для проверки — открытый.
Создадим отсоединённую ЭЦП файла mydocument.pdf в текст. формате:
gpg2 --sign --detach-sign --default-key 29FEB0EF --armor mydocument.pdf
Создадим отсоединённую подпись в двоичном формате:
gpg2 --sign --detach-sign --default-key 29FEB0EF mydocument.pdf
на выходе будет получен файл mydocument.pdf.sig
Создадим встроенную в файл подпись в текстовом формате:
gpg2 --sign --default-key 29FEB0EF --armor mydocument.pdf
Создадим встроенную в файл подпись в двоичном формате:
gpg2 --sign --default-key 29FEB0EF mydocument.pdf
При создании встроенных подписей содержимое файла-источника целиком включается внутрь, поэтому использовать данный формат не желательно из-за дублирования и значительного размера. Поэтому отсоединённая ЭЦП является самым популярным вариантом подписи.

Импорт открытого ключа

Для проверки чужой цифровой подписи GnuPG, у нас должны быть:

1. открытый ключ человека, который её создал;
2. оригинальный файл и файл отсоединённой цифровой подписи.

Сначала импортируем ключ респондента, подписавшего файл (если это не было сделано ранее). Это можно сделать любым способом:

текстовый файл;
серверы-хранилища ключей;
буфер обмена (для GUI утилит).

Импортируем открытый ключ из файла:
gpg2 --import mykey.asc
Здесь mykey.asc — имя файла с открытым ключом.
Теперь мы должны установить доверие импортированному ключу, т.к. в противном случае не сможем проверить подпись. Войдём в интерактивный режим:
pg2 --edit-key 29FEB0EF

Установим доверие ключу:
trust
Проверим отпечаток респондента (например посредством телефонного звонка или любым другим способом), затем выберем пункт **Я полностью доверяю** (***I trust fully***).

Выходим из интерактивного режима:
quit

Проверка ЭЦП

Файл отсоединённой ЭЦП должен лежать в том же каталоге, что и оригинальный файл, иначе выполнить проверку его подлинности будет невозможно.
Проверка отсоединённой подписи файла:
gpg2 --verify mydocument.pdf.sig

Дополнительно

изучите вопросы экспорта (и импорта) ключей и их связок на **ключевые сервера,** реализуйте данные способы экспорта/импорта на выбранный сервер, отобразив результаты в отчете .

По возможности, также, опробуйте встраивание средств GnuPG в какую-либо среду коммуникаций, например, в почтовый клиент Thunderbird или др.

ТРЕБОВАНИЯ К СОСТАВЛЕНИЮ ОТЧЕТА

По результатам выполнения лабораторных работ необходимо составлять отчет. Рекомендуется составлять отчет параллельно с выполнением заданий лабораторных работ, не оставляя этот процесс на потом. Отчет должен быть оформлен с титульным листом с названием университета, института и высшей школы. На титульном листе указывается ф.и.о. студента исполнителя работы, номер студенческой группы и ф.и.о. преподавателя, а также название изучаемого предмета.

В отчете приводится информация, удостоверяющая выполнение студентом заданий лабораторных работ: скриншоты с результатами исполнения сценариев, команд и программ, командных файлов, дампы памяти и пакетов, диаграммы и таблицы, составленные вами скрипты (*BASH* или др.), ответы на поставленные в заданиях вопросы и т.д.

Результаты выполнения каждого из заданий должны предваряться в отчете фрагментом текста, формулирующим само задание (фрагменты могут копироваться прямо из текста заданий).

Защита лабораторных работ происходит по предъявлению оформленного отчета (отчет может быть единым для всего цикла работ, с оглавлением в начале) и сопровождается демонстрацией исполнения программ, сценариев (выборочно из всех заданий) а также ответами на вопросы преподавателя.

Отчет можно создавать под *ОС Linux*, например, с помощью приложения *LibreOffice Writer* , сохраняя в *Home* каталоге вашего рабочего компьютера. Либо можете использовать офисные средства от MS Windows или MacOS.