Время выполнения - одна неделя.

Язык программирования – С.

Для криптографии использовать любую open source криптографическую библиотеку (**mbedtls**, openssl, libsodium, etc.)

Разработать 2 консольных приложения под Linux, клиент и сервер.

**Клиент.**

Имеет 3 параметра командной строки:

*id* - строка с именем клиента, до 8 символов

*delay* - задержка между исходящими сообщениями в ms

*addr* - строка ip:port для подключения к серверу

Алгоритм работы клиента:

1. Генерируется случайный ключ для алгоритма симметричного шифрования (AES, размер ключа на свое усмотрение).

2. Создается пакет данных - *id* клиента (строка), счетчик сообщения (начинается с 0) (строка) и дополняется любым паддингом до 256 байт.

3. Считается хеш сумма (SHA-1) всего пакета данных, и дописывается в конце пакета данных.

4. Пакет данных шифруется симметричным алгоритмом, сгенерированным ранее ключом.

5. Симметричный ключ шифруется ассиметричным шифром (ключи сгенерированы заранее и прописаны в коде программы как константа, алгоритм – RSA, параметры – на свое усмотрение).

6. Зашифрованный симметричный ключ и пакет данных отправляется в едином сообщении по UDP протоколу по адресу указанному в *addr.*

7. Задержка размером *delay* и повторение алгоритма с инкрементом счетчика.

Условия:

Выход из приложения по ctrl-c с выводом количества отправленных сообщений в консоль.

Количество одновременно запущенных приложений не ограничено.

**Сервер.**

Имеет 2 параметра командной строки:

*port* - номер порта, на котором ожидается прием пакетов

*threads* - максимальное количество параллельных потоков обрабатывающих пакет данных

Алгоритм работы сервера:

1. Ожидание входящих пакетов на UDP порту *port.*

2. Обработка входящего соединения и передача пакета на асинхронную дешифровку в отдельный поток.

3. Дешифровка симметричного ключа заранее сохраненным ассиметричным ключом.

4. Дешифровка пакета данных.

5. Проверка хеш суммы.

6. Запись данных без паддинга в единый linked list (в случае ошибки хеш суммы - запись строки "error")

Условия:

Максимальное количество потоков обрабатывающих пакет - не больше *threads*.

При нажатии ctrl-c - сохранение всех данных из linked list в файл и безопасный выход из программы.

Асинхронная обработка сетевых соединений допускается.