Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ОТЧЕТ

О ЛАБОРАТОРНОЙ РАБОТЕ

Лабораторная работа №1

по теме:

«Изучение свойств криптографических функций хеширования»

Исполнитель, студент группы 201-361 \_\_\_\_\_\_\_\_\_\_\_\_ Н.А. Фельдбуш

Москва, 2023

**Постановка задачи**:

Напишите программу, генерирующую из файла leasing.txt эквивалентные по смыслу текстовые документы в количестве, достаточном (условно) для возникновения коллизии функции хеширования SHA-1. Типовые приемы: замена слов и словосочетаний на синонимы; исключение или включение союзов, вводных слов и эпитетов; внедрение управляющих символов. Оцените изменение объема файла после модификации.

**Ход работы:**

Коллизия SHA-1 означает ситуацию, когда два разных входных сообщения дают одинаковое хэш-значение. Это связано с тем, что размер хэша SHA-1 – 160 бит, что позволяет находить коллизии с помощью современных вычислительных мощностей.

Выполним поставленную задачу при помощи средств языка программирования python. Перед началом следует обратить внимание на функцию “init” (рис.1).



Рисунок 1 – Функция “init”

Эта функция проводит инициализацию: создает файл с названием “leasing0.txt” и устанавливает счетчик равным единице.

Затем при запуске файла, мы вызываем функцию “main”, внутри которой проводится инициализация с помощью функции “init” (рис.2).

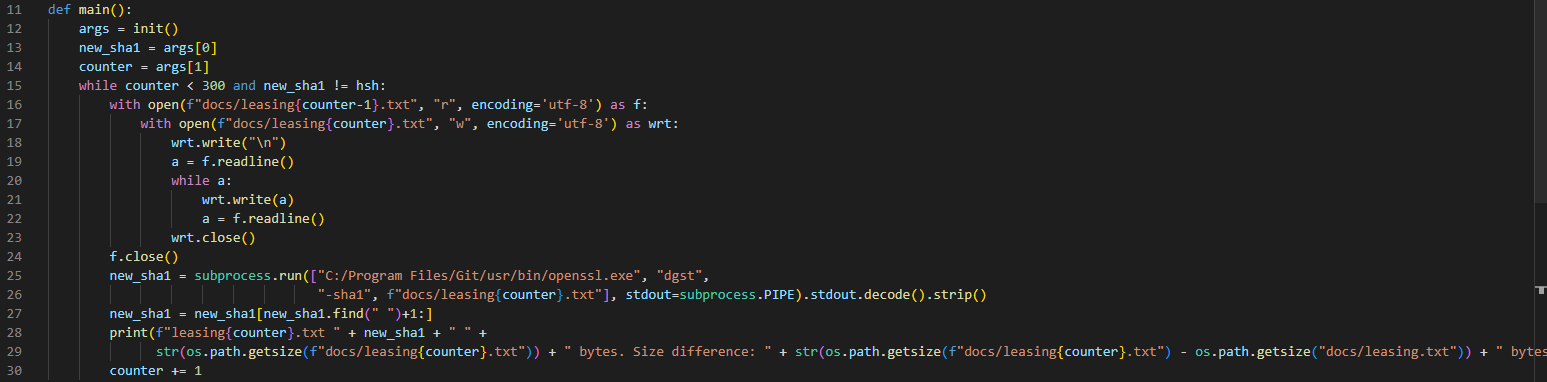


Рисунок 2 – Функция “main”

Внутри функции “main” открывается созданный в “init” файл, в этот файл добавляется символ “\n”, что означает переход на новую строку, а затем создается новый файл с этим добавленным символом.

Вслед за этим с помощью библиотеки “subprocess” происходит получение хеша нового файла, а к счетчику прибавляется единица.

Цикл “while” происходит до тех пор, пока счетчик не станет равным 300, или пока хэш исходного файла не совпадет с хэшем нового. Результатом работы программы является создание от 1 до 300 новых файлов, в зависимости от выполнения второго условия (рис.3).

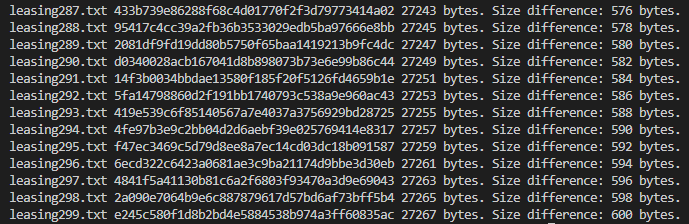


Рисунок 3 – Результат работы программы