**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет прикладной математики-процессов управления**

**Программа бакалавриата**

**“Большие данные и распределенная цифровая платформа”**

**ОТЧЁТ**

**По лабораторной работе №3**

**По дисциплине “Алгоритмы и структуры данных”**

**На тему “Деобезличивание датасета из номеров телефона с солью”**

**Вариант №1**

**Студент гр. 23Б16-пу**

**Сон М.П.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2024 г.**

[**Цель работы 3**](#_jahfja53p111)

[**Описание задачи 3**](#_8kxv256himc6)

[**Теоретическая часть 3**](#_xjfc55o05608)

[**Описание программы 4**](#_udaf87v6f57x)

[**Основные шаги программы 4**](#_16gvlta3yttv)

[**Блок схема программы 5**](#_ewwj3lhgpv52)

[**Рекомендации программиста 7**](#_g5zaia720az2)

[**Контрольный пример 7**](#_rakrum12il8r)

[**Исследование 7**](#_p67uyvwp7lsb)

[**Вывод 8**](#_vqaxqg6rtatf)

# Цель работы

Расшифровать набор данных, зашифрованный с помощью хеш-функции с использованием модификатора входа – соли, а также проанализировать решение аналогичной задачи при различных условиях.

# Описание задачи

* Изучить особенности шифрования телефонных номеров.
* Написать программу для деобезличивания датасета.
* Протестировать программу на выданном варианте.
* Протестировать с еще минимум 2-мя различными хеш-функциями (1 алгоритм на 1 семейство), которыми зашифрован исходный деобезличенный набор.
* Написать от чего меняется скорость расшифровки. Влияние вида соли, длины соли и хеш-функции на скорость расшифровки датасета. А также ответить на вопрос – сколько нужно знать телефонов из предложенного датасета, чтоб 100% его взломать?

# Теоретическая часть

Хеш-функция или функция свёртки — функция, преобразующая массив входных данных произвольного размера в выходную битовую строку определённого (установленного) размера в соответствии с определённым алгоритмом.

Хеш-функция должна обладать следующими свойствами:

* **Определенность (детерминированность)**: Для любых двух одинаковых входных данных хеш-функция всегда должна выдавать одинаковый хеш. Это свойство обеспечивает стабильность результатов.
* **Быстрота вычисления**: Хеш-функция должна эффективно и быстро обрабатывать входные данные, независимо от их размера. Это позволяет использовать хеш-функции в приложениях, требующих высокой производительности.
* **Пред-образная стойкость**: Невозможно (или крайне сложно) найти оригинальные входные данные, зная только их хеш-значение. Это свойство важно для обеспечения безопасности в криптографических приложениях.
* **Стойкость к коллизиям**: Очень сложно найти две различные входные строки, которые дают одинаковое хеш-значение. Это свойство предотвращает случайные или злонамеренные совпадения.
* **Аваланч-эффект**: Небольшое изменение входных данных должно приводить к значительному изменению хеш-значения. Это свойство делает хеш-функции особенно полезными в криптографии и проверке целостности данных.
* **Равномерное распределение**: Хорошая хеш-функция должна распределять хеш-значения равномерно по всему диапазону возможных значений, чтобы избежать кластеризации и уменьшить вероятность коллизий.

Так же хеш-функция может иметь соль - это добавочные данные, которые могут быть однозначно вычислены (или просто заданы) для каждой строки. Они добавляются к изначальному массиву данных, и хеширование происходит для новых данных с солью. Использование соли сильно усложняют расшифровку данных.

Взламывание хеша происходит с помощью техники грубой силы или BruteForce. Это метод при котором происходит хеширование всех возможных вариантов строк, в случае номеров телефона это строки удовлетворяющие маске 8?d?d?d?d?d?d?d?d?d?d и сравнение с уже существующим хешем, при совпадении хеша, строка считается расшифрованной, так как хеш-функция является детерминированной, однако в случае использования соли мы получим не ту строку, которую хотели бы.

# Описание программы

исполняемый файл: lab\_3.py

входной файл с хешами: max\_hashes.txt

Таблица 1. Описание функций

| Название функции | Описание |
| --- | --- |
| personalize | функция расшифровки датасета с номерами телефонов, в качестве параметров необходимо указать выходной файл, входной файл с хешами, и необязательный параметр: тип хеш-функции, которой был зашифрован датасет (тип необходимо взять из документации hashcat), по умолчаний 0 - md5 |
| find\_salt | для расшифрованного датасета, при известных 5 номерах телефонов вычисляет соль |
| write\_right\_phones | записывает в файл right\_phone\_number.txt расшифрованные с учётом соли номера телефонов |
| hash\_md5 | зашифровывает каждую строку переданного файла хеш-функцией md5 и записывает результат в файл md5.txt |
| hash\_sha1 | зашифровывает каждую строку переданного файла хеш-функцией sha1 и записывает результат в файл sha1.txt |
| hash\_sha256 | зашифровывает каждую строку переданного файла хэш-функцией sha256 и записывает результат в файл sha256.txt |

# 

# Основные шаги программы

1. Запуск программы (lab\_3.py)
2. Ожидаем окончания программы

# Блок схема программы

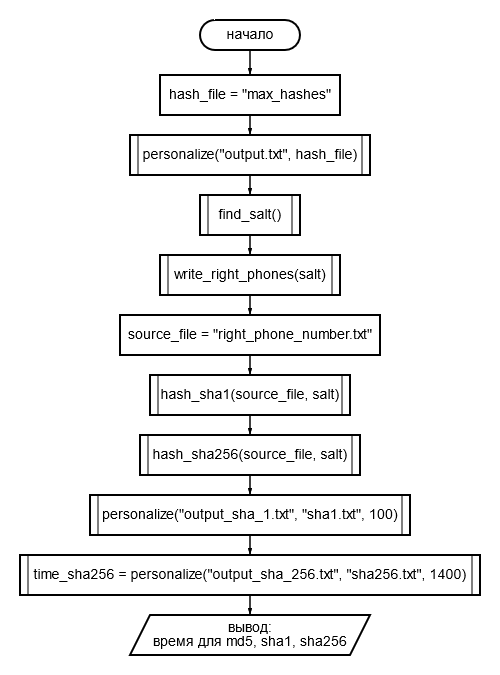


Рис1. Блок схема программы

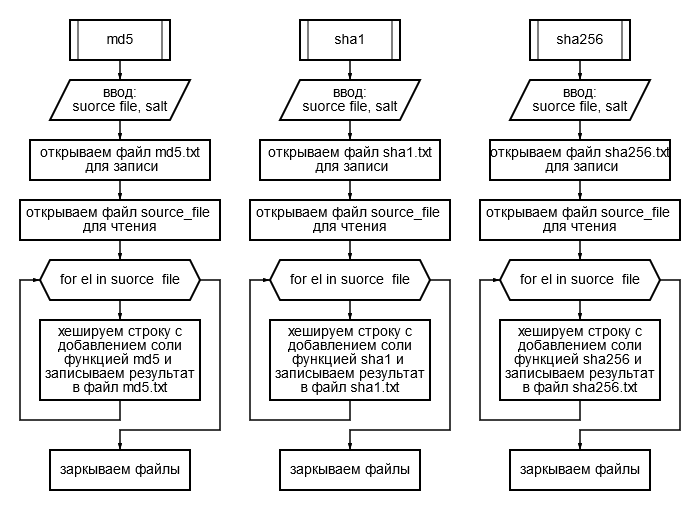
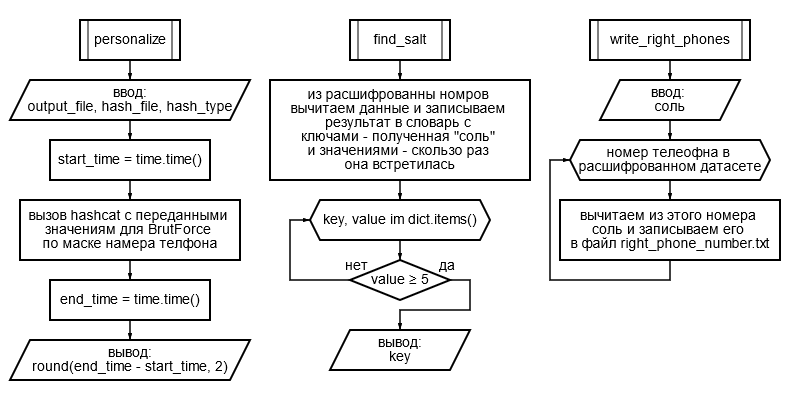
Рис2. Блок-схемы функций хеширования

Рис3. Блок-схемы оставшихся подпрограмм

Рекомендации пользователя

* Для запуска программы, откройте файл lab\_3.py в pycharm и нажмите Shift+F10
* Дождитесь пока на экране не появиться время для расшифровки датасета для разных хеш-функция и одной соли

# Рекомендации программиста

* Используйте python версии не мене 3.11
* Используйте hashcat версии не менее 6.2.5

# Контрольный пример

Запускаем файл lab\_3.py и ждём. В итоге будет выведено время, потребовавшееся для расшфровки данного датасета с одной солью для хеш-функций md5, sha1, sha256

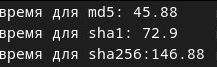


Рис4. Пример вывода

# Исследование

После исполнения программы видно, что скорость расшифровки сильно зависит от функции, которой был зашифрован датасет:

Таблица2. Время расшифровки датасета с одной солью, но разными функциями

| Функция | Время расшифровки |
| --- | --- |
| md5 | 45.88 |
| sha1 | 72.9 |
| sha256 | 146.88 |

Получается, что на расшифровку sha1 тратиться примерно в 1,6 раз больше времени по сравнению с md5, а на sha256 тратится примерно в 2 раза больше времени по сравнению с sha1

Далее было протестировано влияние длины числовой соли, которая была добавлена к номеру телефона, как в данном датасете

Таблица3. Время расшифровки датасета, в зависимости от длины числовой соли

| Длинна соли | Время расшифровки для md5 (сек) | Время расшифровки для sha1 (сек) | Время расшифровки для sha256 (сек) |
| --- | --- | --- | --- |
| 1 | 44.98 | 71.81 | 144.08 |
| 2 | 45.35 | 74.48 | 146.55 |
| 3 | 45.34 | 73.47 | 144.63 |
| 4 | 45.34 | 71.38 | 144.53 |
| 5 | 45.35 | 73.39 | 144.54 |
| 6 | 45.44 | 73.38 | 143.54 |
| 7 | 45.54 | 71.48 | 146.55 |
| 8 | 44.98 | 70.8 | 146.64 |
| 9 | 45.34 | 70.37 | 143.54 |
| 10 | 44.34 | 70.56 | 142.63 |

Итог: на время расшифровки не виляет длина числовой соли, которая складывается с номером телефона

Также было проведено исследование влияние соли, состоящей из маленьких букв английского алфавита и приписываемой справа от номера телефона.

Таблица4. Влияние соли, состоящей из маленьких букв английского алфавита

| Длинна соли | Время расшифровки для md5 | Время расшифровки для sha1 | Время расшифровки для sha256 |
| --- | --- | --- | --- |
| 1 | 17мин 32сек | 28мин | 57мин 30сек |
| 2 | 7час 26мин | 12час 20мин | 1день 1час |
| 3 | 8дней 1час | 13дней 6час | 26денй 16час |

Итог: наличие и длина такой соли сильнее всего влияет на время дешифровки датасета.

# Вывод

В рамках работы было написано программное обеспечение, которое дешифрует датасет, вычисляет соль, шифрует его ещё 2 различными хеш-функциями и снова дешифрует. Также было проведено исследование, на влияние хеш-функции, вида и длины соли на время, необходимое на деобезличивание датасета. В итоге наибольшее влияние оказывает наличие и длина буквенной соли, которая приписывается к номеру телефона, никакого влияния не оказывает наличие и длина числовой соли, складываемой как число с номером телефона, а хеш-функции среди хеш-функций самой тяжелой для деолбезличивания стала sha256, далее sha1 и самой легко-взламываемой является md5.