**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 №5**

**Студент гр. 23Б16-пу**

**Пушкарев Н.П.**

**Преподаватель**

**Дик А.Г.**

**Санкт-Петербург**

**2024 г.**

[**Цель работы 3**](#_ixkrmexdlmx1)

[**Описание задачи 3**](#_2d07uirr35d9)

[**Теоретическая часть 3**](#_bc8ka7j09wvg)

[**Описание программы 4**](#_3znysh7)

[**Основные шаги программы 5**](#_tyjcwt)

[**Блок схема программы 5**](#_3dy6vkm)

[**Рекомендации пользователя 8**](#_jmd7r1jlo5hs)

[**Рекомендации программиста 8**](#_1t3h5sf)

[**Контрольный пример 8**](#_4d34og8)

[**Исследование 8**](#_2s8eyo1)

[**Вывод 9**](#_17dp8vu)

# Цель работы

Цель данной работы – выполнить расшифровку данных, зашифрованных с использованием хеш-функции и уникального модификатора (соли), а также провести анализ решения этой задачи в различных условиях.

# Описание задачи

- Изучить особенности шифрования телефонных номеров.

- Создать программу для деобезличивания датасета.

- Протестировать программу на предоставленном наборе данных.

- Проверить программу с использованием не менее двух различных хеш-функций из разных семейств (по одной для каждого семейства) для зашифрованного датасета.

- Проанализировать, как на скорость расшифровки влияют различные параметры: тип соли, её длина и выбранная хеш-функция. Также выяснить, сколько нужно знать телефонных номеров из датасета для полной расшифровки данных.

# Теоретическая часть

Хеш-функция – это алгоритм, преобразующий массив данных произвольного размера в битовую строку фиксированной длины. Основные свойства хеш-функции включают:

- Детерминированность: одинаковые входные данные всегда дают одинаковый хеш, обеспечивая стабильность результатов.

- Эффективность: хеш-функция должна обрабатывать данные быстро, независимо от их размера, что важно для высокопроизводительных приложений.

-Предобразная стойкость: невозможно или крайне сложно восстановить оригинальные данные, зная только их хеш.

- Стойкость к коллизиям: вероятность того, что разные входные данные дадут одинаковый хеш, должна быть минимальной.

- Аваланч-эффект: небольшое изменение входных данных приводит к значительному изменению хеша, что важно для криптографической стойкости и проверки целостности данных.

- Равномерное распределение: хеш-функция должна равномерно распределять значения, чтобы минимизировать кластеризацию и снизить вероятность коллизий.

Соль представляет собой дополнительные данные, которые добавляются к исходным данным перед их хешированием, что значительно усложняет расшифровку.

Расшифровка хешей, как правило, проводится методом перебора (Brute Force), при котором хешируются все возможные строки, например, строки, соответствующие маске телефонного номера (формат 8XXXXXXXXXX). Сравнивая их хеши с целевыми значениями, можно выявить совпадения и таким образом восстановить исходные данные. Однако при использовании соли восстановить исходную строку затруднительно, поскольку итоговый хеш учитывает и соль.

# 

# 

# 

# 

# 

# 

# 

# Описание программы

исполняемый файл: lab\_3.py

входной файл с хешами: hashes.txt

Таблица 1. Описание функций

| Название функции | Описание |
| --- | --- |
| dehash | Функция для расшифровки датасета с номерами телефонов. Принимает параметры: выходной файл, входной файл с хешами и необязательный параметр – тип хеш-функции, использованной для шифрования данных (тип должен соответствовать форматам, описанным в документации hashcat). По умолчанию используется тип 0 – md5. |
| find\_salt | Вычисляет соль для расшифрованного датасета при условии, что известно 5 номеров телефонов. |
| delete\_salt | Записывает расшифрованные номера телефонов с учетом соли в файл phones.txt. |
| hash\_md5 | Шифрует каждую строку из переданного файла с использованием хеш-функции md5 и записывает результаты в файл md5.txt. |
| hash\_sha1 | Шифрует каждую строку из переданного файла с использованием хеш-функции sha1 и записывает результаты в файл sha1.txt. |
| hash\_sha256 | Шифрует каждую строку из переданного файла с использованием хеш-функции sha256 и записывает результаты в файл sha256.txt. |

# 

# Основные шаги программы

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

# Блок схема программы

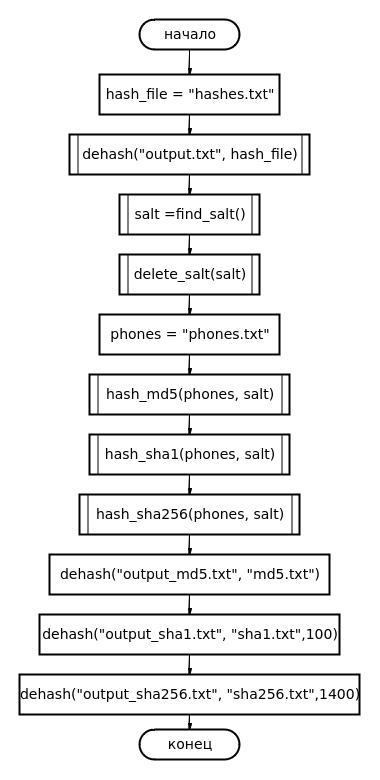


Рис1. Блок схема программы

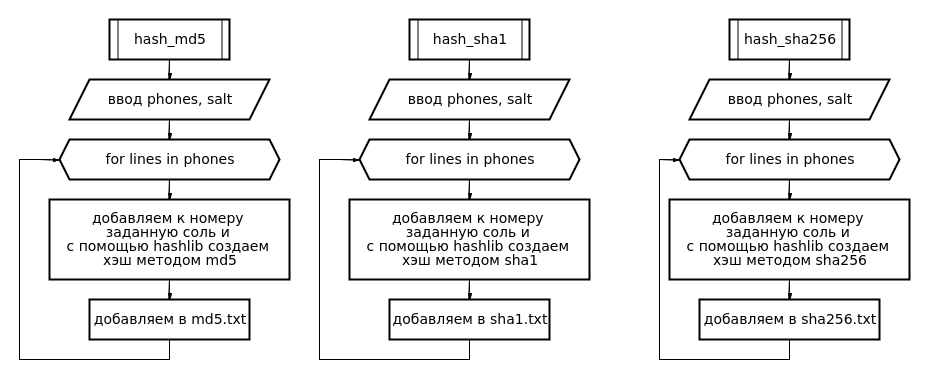


Рис2. Блок-схемы функций хеширования

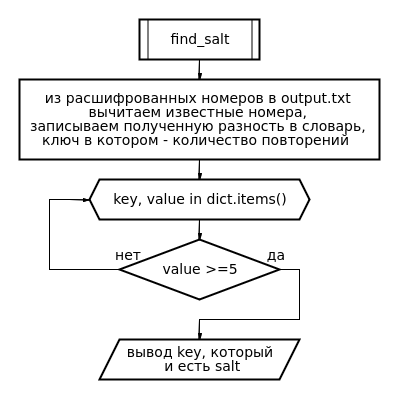
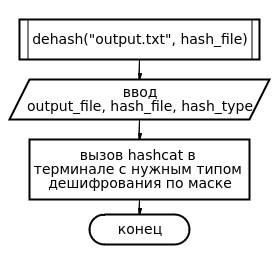


Рис3. Блок-схемы оставшихся подпрограмм

# 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

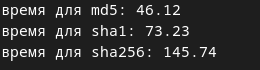


Рис4. Пример вывода

# Исследование

Вывод программы дает понять разницу в скорости расшифровки хэшей, созданных разными методами шифрования:

Таблица2. Время расшифровки датасета разными функциями

| Функция | Время расшифровки |
| --- | --- |
| md5 | 46.12 |
| sha1 | 73.23 |
| sha256 | 145.74 |

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 показаны результаты тестов по дешифрованию с числовой солью разной длины.

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

| Длина соли | Время расшифровки для md5 (сек) | Время расшифровки для sha1 (сек) | Время расшифровки для sha256 (сек) |
| --- | --- | --- | --- |
| 1 | 44.84 | 71.8 | 144.48 |
| 2 | 44.93 | 74.42 | 146.6 |
| 3 | 45.31 | 73.56 | 144.78 |
| 4 | 45.4 | 73.5 | 145.11 |
| 5 | 45.43 | 73.82 | 145.05 |

Можно сделать вывод, что влияние длины соли невелико и не постоянно.

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

| Длина соли | Время расшифровки для md5 | Время расшифровки для sha1 | Время расшифровки для sha256 |
| --- | --- | --- | --- |
| 1 | 19мин 1сек | 28мин | 57мин 30сек |
| 2 | 7час 24мин | 12час 29мин | 1день 1час |

Можно подвести итог, что на время дешифровки датасета сильнее всего влияет буквенная соль.

# Вывод

В рамках работы было разработано программное обеспечение, которое выполняет дешифровку датасета, определяет соль, зашифровывает его с использованием двух других хеш-функций и затем снова расшифровывает. Также было проведено исследование влияния типа хеш-функции, вида и длины соли на время, требуемое для деобезличивания датасета. В результате оказалось, что наибольшее влияние оказывает наличие и длина буквенной соли, добавляемой к номеру телефона. Наличие и длина числовой соли, которая прибавляется к номеру телефона, не оказывают влияния. Среди хеш-функций наиболее трудной для деобезличивания оказалась sha256, за ней следует sha1, а наиболее легко взламываемой является md5.