Ковкель Никита, ФИТ 3-4

Информационная безопасность

Отчет по лабораторной работе № .11

**Цель:** изучение алгоритмов хеширования и приобретение практических навыков их реализации и использования в криптографии.

**Задачи:**

* Закрепить теоретические знания по алгебраическому описанию и алгоритмам реализации операций вычисления однонаправленных хеш-функций.
* Освоить методику оценки криптостойкости хеш-преобразований на основе «парадокса дня рождения».
* Разработать приложение для реализации заданного алгоритма хеширования (из семейств MD и SHA).
* Оценить скорость вычисления кодов хеш-функций.

Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.

1 Практическая часть

1. Разработать оконное приложение, реализующее один из алгоритмов хеширования из указанного преподавателем семейства (MD или SHA; или иного).

Данный код представляет собой простое приложение на Python с графическим интерфейсом, созданное с помощью библиотеки Tkinter. Оно позволяет пользователю ввести текст в поле ввода, выбрать алгоритм хеширования (SHA-1, SHA-256, SHA-384 или SHA-512) из выпадающего списка, после чего по нажатию на кнопку "Хеш" вычисляет хеш-значение введенного текста и выводит его в поле результата, а также отображает время, затраченное на вычисление хеша. Код включает в себя определение функции hash\_text(), которая выполняет вычисление хеша в зависимости от выбранного алгоритма, и создание графического интерфейса с полями ввода, выпадающим списком, кнопкой и полем вывода результата.

|  |
| --- |
| import hashlib  from tkinter import \*  from tkinter import ttk  import time  def hash\_text():  input\_text = input\_field.get()  hash\_algorithm = hash\_algorithm\_combobox.get()    start\_time = time.perf\_counter\_ns()    if hash\_algorithm == "SHA-1":  hash\_value = hashlib.sha1(input\_text.encode()).hexdigest()  elif hash\_algorithm == "SHA-256":  hash\_value = hashlib.sha256(input\_text.encode()).hexdigest()  elif hash\_algorithm == "SHA-384":  hash\_value = hashlib.sha384(input\_text.encode()).hexdigest()  elif hash\_algorithm == "SHA-512":  hash\_value = hashlib.sha512(input\_text.encode()).hexdigest()  else:  hash\_value = "Invalid algorithm selected"    end\_time = time.perf\_counter\_ns()  hash\_time = (end\_time - start\_time) / 1\_000\_000\_000 # Преобразуем в секунды    output\_field.delete(0, END)  output\_field.insert(0, hash\_value)    time\_label.configure(text=f"Время хеширования: {hash\_time:.9f} секунд")  root = Tk()  root.title("SHA Hashing Tool")  main\_frame = ttk.Frame(root, padding=(20, 20))  main\_frame.grid(row=0, column=0, sticky=(N, S, E, W))  input\_label = ttk.Label(main\_frame, text="Введите текст:")  input\_label.grid(row=0, column=0, sticky=W)  input\_field = ttk.Entry(main\_frame)  input\_field.grid(row=1, column=0, sticky=(N, S, E, W))  hash\_algorithm\_label = ttk.Label(main\_frame, text="Алгоритм:")  hash\_algorithm\_label.grid(row=2, column=0, sticky=W)  hash\_algorithm\_combobox = ttk.Combobox(main\_frame, values=["SHA-1", "SHA-256", "SHA-384", "SHA-512"])  hash\_algorithm\_combobox.grid(row=3, column=0, sticky=(N, S, E, W))  hash\_algorithm\_combobox.current(0)  hash\_button = ttk.Button(main\_frame, text="Хеш", command=hash\_text)  hash\_button.grid(row=4, column=0, sticky=(N, S, E, W))  output\_label = ttk.Label(main\_frame, text="Результат:")  output\_label.grid(row=5, column=0, sticky=W)  output\_field = ttk.Entry(main\_frame)  output\_field.grid(row=6, column=0, sticky=(N, S, E, W))  time\_label = ttk.Label(main\_frame, text="")  time\_label.grid(row=7, column=0, sticky=W)  root.mainloop() |

Листинг 11.1 – Код SHA

На данных скриншотах отображен процесс работы программы хэширования алгоритмом семейства SHA-1.

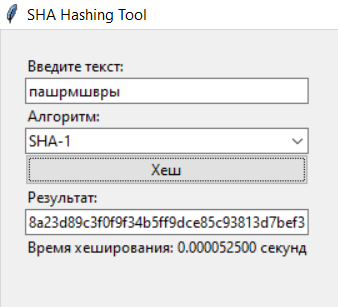


Рисунок 2.1 – Результат хэширования алгоритмом SHA-1.

Алгоритмы SHA (Secure Hash Algorithm) являются чрезвычайно эффективными и быстрыми в вычислении криптографических хешей.

Вывод: в ходе выполнения лабораторной работы были изучены и приобретены практические навыки разработки и использования приложений для реализации алгоритма семейства SHA.