Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Факультет программной инженерии и компьютерных технологий

Отчет по лабораторной работе № 1

“Основы шифрования данных”

по дисциплине Информационная безопасность

Вариант 10

|  |  |
| --- | --- |
| Студент группы № P34151 | Шипулин Павел Андреевич |
| Преподаватель | Маркина Татьяна Анатольевна |

Санкт-Петербург

2024

# Цель работы

Изучить основные принципы шифрования информации, ознакомиться с широко известными алгоритмами шифрования, приобрести навыки их программной реализации.

# Вариант задания

10. Реализовать в программе шифрование и дешифрацию содержимого файла по методу Цезаря. Провести частотный анализ зашифрованного файла, осуществляя проверку по файлу с набором ключевых слов.

# Ход работы

1. Ознакомиться с теоретическими основами шифрования данных.
2. Получить вариант задания у преподавателя.
3. Написать программу согласно варианту задания.
4. Отладить разработанную программу и показать результаты работы программы преподавателю.
5. Составить отчет по лабораторной работе.

# Листинг программ

Ссылка на репозиторий:

<https://github.com/PashcalE2/IS/tree/main/cryptography/first_part>

## Файл lab1.py

import pandas as pd  
  
  
class Alphabet:  
 def \_\_init\_\_(  
 self,  
 first\_lower\_ord: int,  
 first\_upper\_ord: int,  
 size: int  
 ):  
 self.first\_lower\_ord = first\_lower\_ord  
 self.first\_upper\_ord = first\_upper\_ord  
 self.size = size  
  
 self.lowers = [chr(first\_lower\_ord + i) for i in range(size)]  
 self.uppers = [chr(first\_upper\_ord + i) for i in range(size)]  
  
 def is\_lower(self, char: str) -> bool:  
 return char[0] in self.lowers  
   
 def to\_lower(self, s: str) -> str:  
 result = ""  
 for char in s:  
 if self.is\_upper(char):  
 result += self.lowers[ord(char) - self.first\_upper\_ord]  
 else:  
 result += char  
 return result  
  
 def is\_upper(self, char: str) -> bool:  
 return char[0] in self.uppers  
  
 def in\_alphabet(self, char: str) -> bool:  
 return self.is\_lower(char) or self.is\_upper(char)  
  
  
class CaesarMethod:  
 def \_\_init\_\_(self, alphabets: list[Alphabet]):  
 self.alphabets = alphabets  
  
 def rotate(self, s: str, n: int) -> str:  
 result = ""  
 for char in s:  
 not\_rotated = True  
 for alphabet in self.alphabets:  
 if alphabet.is\_lower(char):  
 new\_ord = (ord(char) - alphabet.first\_lower\_ord + n) % alphabet.size  
 result += alphabet.lowers[new\_ord]  
 not\_rotated = False  
 break  
 elif alphabet.is\_upper(char):  
 new\_ord = (ord(char) - alphabet.first\_upper\_ord + n) % alphabet.size  
 result += alphabet.uppers[new\_ord]  
 not\_rotated = False  
 break  
  
 if not\_rotated:  
 result += char  
  
 return result  
  
 def in\_alphabet(self, char: str) -> bool:  
 for alphabet in self.alphabets:  
 if alphabet.in\_alphabet(char):  
 return True  
 return False  
  
  
def freq\_analysis(encrypted: str) -> dict:  
 chars\_freq = {}  
 for i in range(len(encrypted)):  
 char = encrypted[i].lower()  
 if \_caesar\_method.in\_alphabet(char):  
 if char in chars\_freq:  
 chars\_freq[char] += 1  
 else:  
 chars\_freq[char] = 1  
  
 return chars\_freq  
  
  
def find\_key\_words(text: str, key\_words: list[str]):  
 result = []  
 for word in key\_words:  
 if word.lower() in text.lower():  
 result.append(word)  
 return result  
  
  
def hack\_caesar(  
 caesar\_method: CaesarMethod,  
 encrypted: str,  
 key\_words: list[str],  
 max\_rotations: int) -> str:  
 acceptable\_key\_words\_count = len(key\_words) // 2  
  
 for n in range(1, max\_rotations):  
 rotation = -n  
 print(f"Сдвиг на: {rotation}")  
  
 decrypted = caesar\_method.rotate(encrypted, rotation)  
  
 chars\_freq = freq\_analysis(decrypted)  
 df = pd.DataFrame(  
 data=[[  
 key,   
 chars\_freq[key],   
 chars\_freq[key] / sum(chars\_freq.values())  
 ] for key in chars\_freq],  
 columns=["Буква", "Количество", "Вероятность"])  
 df = df.sort\_values("Буква")  
 print(df)  
  
 found\_key\_words = find\_key\_words(decrypted, key\_words)  
 found\_key\_words\_count = len(found\_key\_words)  
 print(f"Найденные ключевые слова ({found\_key\_words\_count}): {found\_key\_words}\n")  
  
 if found\_key\_words\_count >= acceptable\_key\_words\_count:  
 return decrypted  
  
  
\_RU = Alphabet(  
 first\_lower\_ord=ord("а"),  
 first\_upper\_ord=ord("А"),  
 size=ord("я") - ord("а") + 1)  
  
\_EN = Alphabet(  
 first\_lower\_ord=ord("a"),  
 first\_upper\_ord=ord("A"),  
 size=ord("z") - ord("a") + 1)  
  
\_caesar\_method = CaesarMethod([\_RU, \_EN])  
  
\_RU\_key\_words = ["и", "в", "то", "том", "на", "из", "под", "нам", "вам"]  
\_EN\_key\_words = ["and", "in", "that", "on", "from", "of", "the", "a", "you", "we"]  
  
\_key\_words = \_RU\_key\_words + \_EN\_key\_words  
\_max\_rotations = max(\_EN.size, \_RU.size)  
  
  
def lab1(filepath: str, rotates\_count: int):  
 with open(filepath, "r", encoding="UTF-8") as file:  
 filedata = file.read()  
  
 def get\_part\_from\_text(text: str):  
 count = min(200, max(len(encrypted) // 2, 1000))  
 return text[:count // 2] + "\n...\n" + text[-count // 2:]  
  
 encrypted = \_caesar\_method.rotate(filedata, rotates\_count)  
 print(f"Зашифрованный текст:\n{get\_part\_from\_text(encrypted)}")  
 with open("./encrypted.txt", "w", encoding="UTF-8") as file:  
 file.write(encrypted)  
  
 decrypted = hack\_caesar(\_caesar\_method, encrypted, \_key\_words, \_max\_rotations)  
 print(f"Расшифрованный текст:\n{get\_part\_from\_text(decrypted)}")  
 with open("./decrypted.txt", "w", encoding="UTF-8") as file:  
 file.write(decrypted)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 """  
 Вариант 10:  
   
 Реализовать в программе шифрование и дешифрацию  
 содержимого файла по методу Цезаря. Провести частотный анализ  
 зашифрованного файла, осуществляя проверку по файлу с набором  
 ключевых слов.  
 """  
  
 lab1("./input.txt", 5)

# Выполнение

## Результат выполнения программы

Зашифрованный текст:

Хежуча з ужрецчн нтчкрркпчшербтаъ нтщухсеынуттаъ цнцчкс

теыкркта тецумйетнк жацчхаъ ериухнчсуз фунцп

...

jrx. Ymj inwjhynts tk btwp ts

ymj hwjfynts tk NU htwjx ktw xdxyjrx-ts-f-hmnu nx wfuniqd ijajqtunsl.

Сдвиг на: -1

Буква Количество Вероятность

44 a 8 0.005161

39 c 18 0.011613

37 e 41 0.026452

43 f 9 0.005806

49 g 19 0.012258

41 h 32 0.020645

34 i 75 0.048387

30 j 19 0.012258

47 k 19 0.012258

36 l 22 0.014194

46 m 51 0.032903

45 o 6 0.003871

48 p 22 0.014194

33 q 30 0.019355

38 r 45 0.029032

31 s 60 0.038710

42 t 13 0.008387

32 v 45 0.029032

40 w 53 0.034194

35 x 60 0.038710

50 y 10 0.006452

51 z 5 0.003226

14 а 5 0.003226

24 в 4 0.002581

27 г 11 0.007097

1 д 77 0.049677

2 е 16 0.010323

6 ж 34 0.021935

21 з 8 0.005161

20 и 19 0.012258

11 й 81 0.052258

28 к 6 0.003871

19 л 15 0.009677

9 м 87 0.056129

23 н 11 0.007097

12 о 29 0.018710

7 п 31 0.020000

17 р 32 0.020645

10 с 74 0.047742

3 т 67 0.043226

22 у 13 0.008387

0 ф 54 0.034839

8 х 50 0.032258

4 ц 54 0.034839

13 ч 17 0.010968

16 ш 8 0.005161

15 щ 26 0.016774

18 ъ 6 0.003871

25 ы 13 0.008387

29 ь 2 0.001290

26 э 2 0.001290

5 я 36 0.023226

Найденные ключевые слова (3): ['и', 'в', 'a']

Сдвиг на: -2

Буква Количество Вероятность

39 b 18 0.011613

37 d 41 0.026452

43 e 9 0.005806

49 f 19 0.012258

41 g 32 0.020645

34 h 75 0.048387

30 i 19 0.012258

47 j 19 0.012258

36 k 22 0.014194

46 l 51 0.032903

45 n 6 0.003871

48 o 22 0.014194

33 p 30 0.019355

38 q 45 0.029032

31 r 60 0.038710

42 s 13 0.008387

32 u 45 0.029032

40 v 53 0.034194

35 w 60 0.038710

50 x 10 0.006452

51 y 5 0.003226

44 z 8 0.005161

24 б 4 0.002581

27 в 11 0.007097

1 г 77 0.049677

2 д 16 0.010323

6 е 34 0.021935

21 ж 8 0.005161

20 з 19 0.012258

11 и 81 0.052258

28 й 6 0.003871

19 к 15 0.009677

9 л 87 0.056129

23 м 11 0.007097

12 н 29 0.018710

7 о 31 0.020000

17 п 32 0.020645

10 р 74 0.047742

3 с 67 0.043226

22 т 13 0.008387

0 у 54 0.034839

8 ф 50 0.032258

4 х 54 0.034839

13 ц 17 0.010968

16 ч 8 0.005161

15 ш 26 0.016774

18 щ 6 0.003871

25 ъ 13 0.008387

29 ы 2 0.001290

26 ь 2 0.001290

5 ю 36 0.023226

14 я 5 0.003226

Найденные ключевые слова (3): ['и', 'в', 'из']

Сдвиг на: -3

Буква Количество Вероятность

39 a 18 0.011613

37 c 41 0.026452

43 d 9 0.005806

49 e 19 0.012258

41 f 32 0.020645

34 g 75 0.048387

30 h 19 0.012258

47 i 19 0.012258

36 j 22 0.014194

46 k 51 0.032903

45 m 6 0.003871

48 n 22 0.014194

33 o 30 0.019355

38 p 45 0.029032

31 q 60 0.038710

42 r 13 0.008387

32 t 45 0.029032

40 u 53 0.034194

35 v 60 0.038710

50 w 10 0.006452

51 x 5 0.003226

44 y 8 0.005161

24 а 4 0.002581

27 б 11 0.007097

1 в 77 0.049677

2 г 16 0.010323

6 д 34 0.021935

21 е 8 0.005161

20 ж 19 0.012258

11 з 81 0.052258

28 и 6 0.003871

19 й 15 0.009677

9 к 87 0.056129

23 л 11 0.007097

12 м 29 0.018710

7 н 31 0.020000

17 о 32 0.020645

10 п 74 0.047742

3 р 67 0.043226

22 с 13 0.008387

0 т 54 0.034839

8 у 50 0.032258

4 ф 54 0.034839

13 х 17 0.010968

16 ц 8 0.005161

15 ч 26 0.016774

18 ш 6 0.003871

25 щ 13 0.008387

29 ъ 2 0.001290

26 ы 2 0.001290

5 э 36 0.023226

14 ю 5 0.003226

Найденные ключевые слова (5): ['и', 'в', 'то', 'из', 'a']

Сдвиг на: -4

Буква Количество Вероятность

37 b 41 0.026452

43 c 9 0.005806

49 d 19 0.012258

41 e 32 0.020645

34 f 75 0.048387

30 g 19 0.012258

47 h 19 0.012258

36 i 22 0.014194

46 j 51 0.032903

45 l 6 0.003871

48 m 22 0.014194

33 n 30 0.019355

38 o 45 0.029032

31 p 60 0.038710

42 q 13 0.008387

32 s 45 0.029032

40 t 53 0.034194

35 u 60 0.038710

50 v 10 0.006452

51 w 5 0.003226

44 x 8 0.005161

39 z 18 0.011613

27 а 11 0.007097

1 б 77 0.049677

2 в 16 0.010323

6 г 34 0.021935

21 д 8 0.005161

20 е 19 0.012258

11 ж 81 0.052258

28 з 6 0.003871

19 и 15 0.009677

9 й 87 0.056129

23 к 11 0.007097

12 л 29 0.018710

7 м 31 0.020000

17 н 32 0.020645

10 о 74 0.047742

3 п 67 0.043226

22 р 13 0.008387

0 с 54 0.034839

8 т 50 0.032258

4 у 54 0.034839

13 ф 17 0.010968

16 х 8 0.005161

15 ц 26 0.016774

18 ч 6 0.003871

25 ш 13 0.008387

29 щ 2 0.001290

26 ъ 2 0.001290

5 ь 36 0.023226

14 э 5 0.003226

24 я 4 0.002581

Найденные ключевые слова (4): ['и', 'в', 'то', 'of']

Сдвиг на: -5

Буква Количество Вероятность

37 a 41 0.026452

43 b 9 0.005806

49 c 19 0.012258

41 d 32 0.020645

34 e 75 0.048387

30 f 19 0.012258

47 g 19 0.012258

36 h 22 0.014194

46 i 51 0.032903

45 k 6 0.003871

48 l 22 0.014194

33 m 30 0.019355

38 n 45 0.029032

31 o 60 0.038710

42 p 13 0.008387

32 r 45 0.029032

40 s 53 0.034194

35 t 60 0.038710

50 u 10 0.006452

51 v 5 0.003226

44 w 8 0.005161

39 y 18 0.011613

1 а 77 0.049677

2 б 16 0.010323

6 в 34 0.021935

21 г 8 0.005161

20 д 19 0.012258

11 е 81 0.052258

28 ж 6 0.003871

19 з 15 0.009677

9 и 87 0.056129

23 й 11 0.007097

12 к 29 0.018710

7 л 31 0.020000

17 м 32 0.020645

10 н 74 0.047742

3 о 67 0.043226

22 п 13 0.008387

0 р 54 0.034839

8 с 50 0.032258

4 т 54 0.034839

13 у 17 0.010968

16 ф 8 0.005161

15 х 26 0.016774

18 ц 6 0.003871

25 ч 13 0.008387

29 ш 2 0.001290

26 щ 2 0.001290

5 ы 36 0.023226

14 ь 5 0.003226

24 ю 4 0.002581

27 я 11 0.007097

Найденные ключевые слова (12): ['и', 'в', 'то', 'том', 'на', 'из', 'and', 'in', 'on', 'of', 'the', 'a']

Расшифрованный текст:

Работы в области интеллектуальных информационных систем

нацелены насоздание быстрых алгоритмов поиск

...

ems. The direction of work on

the creation of IP cores for systems-on-a-chip is rapidly developing.

# Вывод

Изучил основные принципы шифрования, ознакомился с шифром Цезаря и сделал программную реализацию шифра. Этот шифр является одним из самых простых шифров. Найти ключ шифрации (количество ротаций) можно последовательным перебором значений ключа и применением алгоритма дешифрации, поэтому этот шифр не обеспечивает надежную защиту исходного текста.