

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ
НАПРАВЛЕНИЕ ПРОГРАММНАЯ ИНЖЕНЕРИЯ
ОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА СИСТЕМНОЕ И ПРИКЛАДНОЕ
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
СПЕЦИАЛИЗАЦИЯ СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1.1
курса «Информационная безопасность»
по теме: «Криптографические системы с секретным ключом»
Вариант № 4

Выполнил студент:
Тюрин Иван Николаевич
группа: Р33102

Преподаватель:
Маркина Т.А.,
Рыбаков С.Д.

Санкт-Петербург, 2024 г.

Содержание

Лабораторная работа № 1.1. Криптографические системы с секретным ключом	2
1. Описание	2
2. Выполнение задания	2
3. Вывод	6

Лабораторная работа № 1.1

Криптографические системы с секретным ключом

1. Описание

Лабораторная работа № 1 «Основы шифрования данных».

Цель работы. Изучение основных принципов шифрования информации, знакомство с широко известными алгоритмами шифрования, приобретение навыков их программной реализации.

Задание варианта № 4:

4. Реализовать в программе шифрование и дешифрацию файла с использованием квадрата Полибия, обеспечив его случайное заполнение.

2. Выполнение задания

В соответствии с условием задания разработан скрипт на языке Python выполняющий шифрование текста из файла, имя которого подается на ввод, по методу «Квадрата Полибия». Исходный код скрипта можно видеть на листинге [1.1](#).

Утилита принимает на вход 4 аргумента:

1. тип операции (encrypt/decrypt),
2. входной файл,
3. выходной файл,
4. название файла с содержанием «ключа» (значения квадрата Полибия).

При шифровании создается случайный ключ и записывается в файл указанный 3 аргументом.

Пример работы утилиты по шифрованию и дешифрованию фрагмента текста из трактата «Lorem ipsum» можно видеть на изображении [1.1](#).

```

D:\Projects\itmo-info-sec\crypto\lab-1_1> cat original.txt
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in
culpa qui officia deserunt mollit anim id est laborum.

D:\Projects\itmo-info-sec\crypto\lab-1_1> python main.py encrypt original.txt encrypted.txt square.txt; cat encrypted.txt
2214115231??5442154331??4514221411??155423??25315223??4414351552442352234311??25455442541544543553??52225423??155245??4514??5254
4315311445??235231421411??5435445445445433523??4323??222521141152??5223??451422141152??3125533525??252254124325??4323??52355431??
2545??3154355431??135235542531??12435415??35141523114345??525152114454232523541435??43222225314414??22252114115415??35541554??4323
??25225412435442??5251??5225??44143131144514??441435155212432523??445435415??25432352??5411431152??4514221411??5435??11524211525552
354552115423??5435??131422434223252352??1352225423??52151552??44542224331??451422141152??5243??344353542523??354322225??4225115425
234311??52514454223524311??15543523??1444442552442523??4443425445232523??351435??421114544523523??15433523??5435??4443224225
??124354??14343454445425??4552155211433523??31142225423??25355431??5445??521523??22252114114331????
D:\Projects\itmo-info-sec\crypto\lab-1_1> python main.py decrypt encrypted.txt decrypted.txt square.txt; cat decrypted.txt
LOREM?IPSUM?DOLOR?SIT?AMET??CONSECTETUR?ADIPISCING?ELIT??SED?DO?EIUSMOD?TEMPOR?INCIDIDUNT?UT?LABORE?ET?DOLORE?MAGNA?ALIQUA??UT?ENIM?
AD?MINIM?VENIAM??QUIS?NOSTRUD?EXERCITATION?ULLAMCO?LABORIS?NISI?UT?ALIQUIP?EX?EA?COMMODO?CONSEQUAT??DUIS?AUTE?IRURE?DOLOR?IN?REPREHE
NDERIT?IN?VOLUPTATE?VELIT?ESSE?CILLUM?DOLORE?EU?FUGIAT?NULLA?PARIATUR??EXCEPTEUR?SINT?OCCAECAT?CUPIDATAT?NON?PROIDENT??SUNT?IN?CULPA
?QUI?OFFICIA?DESERUNT?MOLLIT?ANIM?ID?EST?LABORUM???
D:\Projects\itmo-info-sec\crypto\lab-1_1> cat square.txt
R:1,1
Q:1,2
V:1,3
O:1,4
S:1,5
B:2,1
L:2,2
T:2,3
W:2,4
A:2,5
M:3,1
Y:3,2
K:3,3
F:3,4
N:3,5
Z:4,1
P:4,2
U:4,3
C:4,4
D:4,5
X:5,1
E:5,2
G:5,3
I:5,4
H:5,5
D:\Projects\itmo-info-sec\crypto\lab-1_1>

```

Рис. 1.1: Пример работы утилиты для шифрования фрагмента трактата Lorem ipsum.

```

1 #!/usr/bin/env python
2
3 # 4. Реализовать в программе шифрование и дешифрацию
4   файла с использованием
5   квадрата Полибия, обеспечив его случайное заполн
6   ение.
7
8 import random
9 import string
10 from typing import Literal
11 from sys import argv
12
13 type PolybiySquare = dict[str, tuple[int, int]]
14 type Mode = Literal["encrypt", "decrypt"]

```

```

14
15 def generate_polybiy_square() -> PolybiySquare:
16     alphabet = string.ascii_uppercase.replace("J", "")
17     assert len(alphabet) == 25, "alphabet length
    shoud be square number"
18
19     permutation = random.sample(alphabet, len(
    alphabet))
20     square = {permutation[i]: (i // 5 + 1, i % 5 + 1)
    for i in range(len(permutation))}
21     return square
22
23
24 def encrypt(text: str, square: PolybiySquare) -> str:
25     text = text.upper().replace("J", "I")
26     encrypted_text = []
27     for char in text:
28         if char in square:
29             row, col = square[char]
30             encrypted_text.append(f"{row}{col}")
31         else:
32             encrypted_text.append("??")
33     return "".join(encrypted_text)
34
35
36 def decrypt(encrypted_text: str, square:
    PolybiySquare) -> str:
37     reverse_square = {v: k for k, v in square.items()
    }
38     decrypted_text = []
39     for i in range(0, len(encrypted_text), 2):
40         try:
41             row, col = int(encrypted_text[i]), int(
    encrypted_text[i + 1])
42             decrypted_text.append(reverse_square[(row
    , col)])
43         except ValueError:
44             decrypted_text.append("?")
45     return "".join(decrypted_text)
46
47
48 def process_file(

```

```

49     input_filename, output_filename, square:
PolybiySquare, mode: Mode = "encrypt"
50 ):
51     with open(input_filename, "r") as file:
52         text = file.read()
53
54     if mode == "encrypt":
55         result = encrypt(text, square)
56     elif mode == "decrypt":
57         result = decrypt(text, square)
58
59     with open(output_filename, "w") as file:
60         file.write(result)
61
62
63 def main(
64     input_filename: str, output_filename: str,
65     square_data_filename: str, mode: Mode
66 ):
67     if mode == "encrypt":
68         square = generate_polibiy_square()
69         with open(square_data_filename, "w") as file:
70             data = "\n".join([f"{k}:{square[k][0]},{
square[k][1]}" for k in square])
71             file.write(data)
72     else:
73         with open(square_data_filename, "r") as file:
74             square = {}
75             for row in file.readlines():
76                 [char, pos] = row.split(":")
77                 [i, j] = map(int, pos.split(","))
78                 square[char] = (i, j)
79
80     process_file(input_filename, output_filename,
81                 square, mode=mode)
82
83 if __name__ == "__main__":
84     if argv[1] == "encrypt":
85         mode = "encrypt"
86     elif argv[1] == "decrypt":
87         mode = "decrypt"
88     else:

```

```
88         raise KeyError("first argument specify wether
89         'encrypt' or 'decrypt' text")
90
91     main(
92         input_filename=argv[2],
93         output_filename=argv[3],
94         square_data_filename=argv[4],
95         mode=mode,
```

Листинг 1.1: Код скрипта выполняющего кодирование и декодирование текста методом «квадрата Полибия»

3. Вывод

В результате выполнения работы были изучены методы шифрования с секретным ключом, в частности с помощью квадрата Полибия.