Лабораторная работа № 7

Дисциплина: Информационная безопасность

Сулицкий Богдан Романович

Содержание

Список иллюстраций

1 Цель работы

Целью данной лабораторной работы является освоение на практике применения режима однократного гаммирования.

2 Выполнение лабораторной работы

1. Я добавил нужные библиотеки для дальнейших действий(2.1).

```
import random
import string
```

Рис. 2.1: Добавление библиотек

2. Я создал класс VernamCipher, принимающий в конструкторе текст как переменную(2.2).

```
class VernamCipher:
    def __init__(self, t, key=None):
        self.P = t
        self.len = len(t)
        self.alf = "абвгдеёжзийклмнопрстуфхцчшщъыьэюя" + string.ascii_lowercase + string.digits
        if key is None:
            self.K = self.key_create()
        else:
            self.K = key
        self.C = self.coder(self.P, self.K)
```

Рис. 2.2: Класс и конструктор

3. Я создал метод, генерирующий ключ шифрования(2.3).

```
def key_create(self):
    return "".join(random.choice(self.alf) for i in range(self.len))
```

Рис. 2.3: Метод генерации ключа шифрования

4. Я создал метод, который не только шифрует, но и дешифрует текст(2.4).

```
def coder(self, line, key):
    return "".join(chr(ord(c) ^ ord(k)) for c, k in zip(line, key))
```

Рис. 2.4: Метод шифровки/дешифровки текста

5. Я создал метод, определяющий ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста(2.5).

```
def find_plaintext(self, fragment):
    keyLen = len(fragment)
    possible_keys = []
    for i in range(len(self.C) - keyLen + 1):
        key = [chr(ord(c) ^ ord(k)) for c, k in zip(self.C[i:i + keyLen], fragment)]
        intact_plaintext = "".join(chr(ord(c) ^ ord(k)) for c, k in zip(self.C, key))
        if fragment in intact_plaintext:
            possible_keys.append(''.join(key))
    return possible_keys
```

Рис. 2.5: Метод определения ключа

6. Я создал инициализацию класса с вводом текста и вызов всех методов класса с последующим выводом данных (2.6).

```
if __name__ == "__main__":
    text = VernamCipher(input("Введите открытый текст: "))
    print("Текст:", text.P,
          "\nКлюч:", text.K,
          "\nШифротекст:", text.C)
    print("Декодированный текст:", text.coder(text.K, text.C))
    print("\nВозможные ключи: ", text.find_plaintext(input("Введите фрагмент открытого текста: ")))
```

Рис. 2.6: Инициализация класса и вывод

7. Результат (2.7)

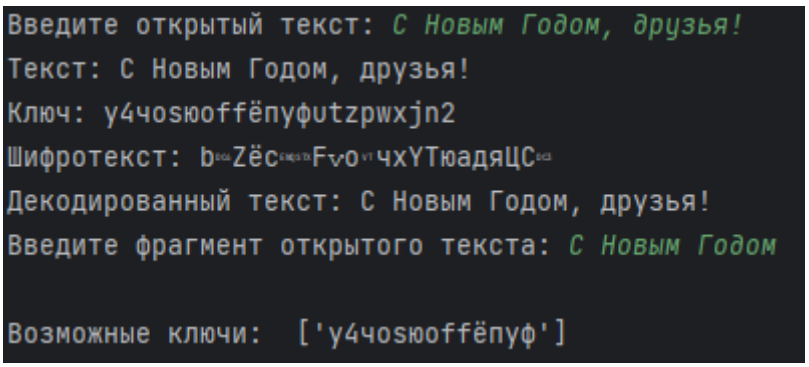

Рис. 2.7: Результат работы программы

3 Вывод

В ходе проделанной лабораторной работы я освоил на практике применение режима однократного гаммирования.

Список литературы

[1] https://esystem.rudn.ru/pluginfile.php/2090284/mod_resource/content/2/007-lab_crypto-gamma.pdf