

Лабораторная работа №7

Основы информационной безопасности

Иванов Сергей Владимирович, НПИбд-01-23

8 мая 2025

Российский университет дружбы народов, Москва, Россия

Освоить на практике применение режима однократного гаммирования.

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

Выполнение работы

Функция генерации ключа

Для начала напишем функцию для генерации случайного ключа (рис. 1)

```
1 import random
2 import string
3
4 # генерация ключа
5 usage
6
7 def generate_key(text):
8     key = ''
9     for i in range(len(text)):
10         key += random.choice(string.ascii_letters + string.digits)
11     return key
```

Рис. 1: Функция генерации ключа

Функция шифрования и дешифрования

Пишу функцию для шифрования и дешифрования текста. (рис. 2).

```
11 # шифрование и дешифрование текста
    3 usages
12 def en_de_crypt(text, key):
13     new_text = ''
14     for i in range(len(text)):
15         new_text += chr(ord(text[i]) ^ ord(key[i % len(key)]))
16     return new_text
```

Рис. 2: Функция шифрования и дешифрования

Пишу функцию для поиска возможных ключей для фрагмента текста (рис. 3).

```
18 # функция для нахождения возможных ключей
19 1 usage
20 def find_keys(text, fragment):
21     possible_keys = []
22     for i in range(len(text) - len(fragment) + 1):
23         possible_key = ''
24         for j in range(len(fragment)):
25             possible_key += chr(ord(text[i + j]) ^ ord(fragment[j]))
26         possible_keys.append(possible_key)
27     return possible_keys
```

Рис. 3: Поиск возможных ключей

Проверка работы программы

Проверяем всех функций. Убеждаемся, все работает корректно. (рис. 4).

```
28 text = 'С новым годом, друзья!'
29 key = generate_key(text)
30 en_crypt = en_de_crypt(text, key)
31 de_crypt = en_de_crypt(en_crypt, key)
32 find_keys_text = find_keys(en_crypt, fragment='С новым ')
33
34 print("Открытый текст:", text, "\nКлюч:", key, "\nШифротекст:", en_crypt, "\nИсходный текст:", de_crypt)
35 print("Возможные ключи:", find_keys_text)
36 print("Расшифрованный фрагмент:", en_de_crypt(en_crypt, find_keys_text[0]))
```

Run main x

C:\Users\lserg\PycharmProjects\Lab07_OIB\.venv\Scripts\python.exe C:\Users\lserg\PycharmProjects\Lab07_OIB\main.py

Открытый текст: С новым годом, друзья!

Ключ: aEv6Bm8KRw05fstZ8j\flLe

Шифротекст: реуJвЦЕкщбНьТзqшbьfD

Исходный текст: С новым годом, друзья!

Возможные ключи: ['aEv6Bm8K', 'ф5N\x140ic', 'jшM\x186P]_', ')\a\x1b:ь*иФ', 'QI9sS\x028Ы', '\x070i_{07о', '%K\\w6@f\x7f', 'ьс

Расшифрованный фрагмент: С новым ЁКони2LХЙи3Мс)

Рис. 4: Проверка работы программы

Вывод

В ходе выполнения лабораторной работы мной было освоено на практике применение режима однократного гаммирования.