

Лабораторная работа №7

Элементы криптографии. Однократное гаммирование


Гудиева Мадина Куйраевна

Цель работы

Освоить на практике применение режима однократного гаммирования.

1. Выполнение лабораторной работы
2. Оформление отчета и презентации
3. Выгрузка видео на youtube и файлов на GitHub

Блок программы с библиотеками и функциями



The screenshot shows a Jupyter Notebook interface with three code cells. The first cell imports the random and string modules. The second cell defines two functions: generate_key, which generates a key of a given size using random characters, and convert_hexadecimal_form, which converts a string to hexadecimal. The third cell defines the use_single_gaming function, which takes a string and a key as input and returns a single gaming result. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution.

```
In [1]: 1 import random
        2 import string

In [2]: 1 def generate_key(size = 6, chars = string.ascii_letters + string.digits):
        2     return ''.join(random.choice(chars) for _ in range(size))
        3 def convert_hexadecimal_form(s):
        4     return "- ".join("{:02x}".format(ord(c)) for c in s)

In [3]: 1 def use_single_gaming(f_string, s_string):
        2     ascii_f_string = [ord(i) for i in f_string]
        3     ascii_s_string = [ord(i) for i in s_string]
        4     return ''.join(chr(s ^ k) for s, k in zip(ascii_f_string, ascii_s_string))
```

Figure 1: Блок программы с библиотеками и функциями

Первый пункт

Определить вид шифротекста при известном ключе и известном открытом тексте.

```
Задание 1

In [4]: 1 cur_string = "С Новым годом, друзья!"
        2
        3 key = generate_key(len(cur_string))
        4
        5 print("Ключ: ", key)
        6 print("Ключ(16): ", convert_hexadecimal_form(key))
        7
        8 new_string = use_single_gamming(cur_string, key)
        9
        10 print("Зашифрованная строка: ", new_string)
        11 print("Зашифрованная строка(16): ", convert_hexadecimal_form(new_string))

Ключ: .yKt17J7ntJMcFRTHHBSa
Ключ(16): 79 57 74 69 37 64 37 6e 74 5a 57 43 66 52 49 48 48 38 42 73 61
Зашифрованная строка: jм15СНмАКтЪм-10jовH8
Зашифрованная строка(16): 458 77 469 457 405 421 40b 4e 467 464 463 47d 45a 7e 69 47c 488 47b 475 43f 42e
```

Figure 2: Блок программы с выполнением первого пункта

Второй пункт

Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

```
Задание 2
In [7]: 1 key = generate_key(len(cur_string))
        2
        3 print("Ключ: ", key)
        4 print("Ключ(16): ", convert_hexadecimal_form(key))
        5
        6 print("Полученная строка: ", use_single_gamming(new_string, key))
        7
        8 key = use_single_gamming(cur_string, new_string)
        9 print("Ключ: ", key)
       10 print("Ключ(16): ", convert_hexadecimal_form(key))
       11
       12 print("Полученная строка: ", use_single_gamming(new_string, key))

Ключ:  S8P01ZM8BXAggCJ5esJF
Ключ(16):  53 38 50 64 49 32 58 57 30 38 58 41 67 71 43 4a 53 65 73 6a 46
Полученная строка:  h0WruF0yKmc0*ah01sK

Ключ:  yWti7Jnt2uCFRiH88sa
Ключ(16):  79 57 74 69 37 6a 37 6e 74 5a 57 43 66 52 49 48 48 38 42 73 61
Полученная строка:  С Новым Годом, друзья
```

Figure 3: Блок программы с выполнением первого пункта

Выводы

Таким образом я успешно освоила на практике применение режима однократного гаммирования.