Лабораторная работа № 7

Элементы криптографии. Однократное гаммирование

Миленин Иван Витальевич

Цель работы

Освоить на практике применение режима однократного гаммирования.

Теоретическое описание

Простейшей и в то же время наиболее надёжной из всех схем шифрования является так называемая схема однократного использования, изобретение, которое чаще всего связывают с именем Г.С. Вернама.

Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, т.е. последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. С точки зрения теории криптоанализа, метод шифрования случайной

Ход работы

1. В первую очередь оговоримся, что использовать будем среду Jupyter Notebook и язык программирования Питон. Для выполнения задания нам необходимо будет подключить библиотеки random и string. Пишем блок необходимых функций, которые и реализуют всю логику программы: функция generate_new_key принимает на вход длину требуемого ключа и возвращает случайную строку символов, что и будет являться ключом; функция hexadecimal_form возвращает шестнадцатиричный вид подаваемой на вход строки; функции single_gamming, unencrypt и compute_initial_key принимают на вход две строки и выполняют их посимвольное сложение по модулю 2 (иллюстр. [-@fig:001]).

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

def generate_new_key(size = 6, chars = string.ascii_letters + string.digits):
    return ''.join(random.choice(chars) for _ in range(size))
def hexadecimal_form(s):
    return " ".join("{:02x}".format(ord(c)) for c in s)

def single_gamming(initial_string, key):
    initial_string_ascii = [ord(i) for i in initial_string]
    key_ascii = [ord(i) for i in key]
    encrypted_string = ''.join(chr(s ^ k) for s, k in zip(initial_string_ascii, key_ascii))
    return encrypted_string
def unencrypt(encrypted_string, key):
    encrypted_string_ascii = [ord(i) for i in encrypted_string]
    key_ascii = [ord(i) for i in key]
    initial_string = ''.join(chr(s ^ k) for s, k in zip(encrypted_string_ascii, key_ascii))
    return initial_string
def compute_initial_key(initial_string, encrypted_string):
    initial_string_ascii = [ord(i) for i in initial_string]
    encrypted_string_ascii = [ord(i) for i in encrypted_string]
    initial_key = ''.join(chr(s ^ k) for s, k in zip(initial_string_ascii, encrypted_string_ascii))
    return initial_key
```

Блок функций для расчетов

2. Далее пишем блок расчетов всех необходимых параметров: initial_string получает с клавиатуры входную строку (“С Новым Годом, друзья!” в нашем случае); key - начальный используемый ключ; encrypted_string - первоначальный шифротекст; new_key - случайный ключ, используемый для получения варианта шифротекста; unencrypted_new_key - вариант текста, получаемый с помощью случайного ключа и изначального шифротекста; initial_key - начальный ключ, получаемый гаммированием открытого начального текста имеющимся шифротекстом; unencrypted_initial_key - расшифрованный вычисленным исходным ключом шифротекст (иллюстр. [-@fig:002]).

In [2]:

```
initial_string = input("Введите начальную строку\n>> ")

key = generate_new_key(len(initial_string))
print("\nИспользуемый ключ:\n", key)
print("В шестнадцатеричном виде:\n", hexadecimal_form(key))

encrypted_string = single_gamming(initial_string, key)

new_key = generate_new_key(len(encrypted_string))
unencrypted_new_key = unencrypt(encrypted_string, new_key)
initial_key = compute_initial_key(initial_string, encrypted_string)
unencrypted_initial_key = unencrypt(encrypted_string, initial_key)
```

```
Введите начальную строку
>> С Новым Годом, друзья!

Используемый ключ:
 wNyjpNeyZlhNAQ5mNjwEjP
В шестнадцатеричном виде:
 77 4e 79 6a 70 4e 65 79 5a 6c 68 4e 41 51 35 6d 4e 6a 77 45 6a 50
```

Блок расчетов переменных

3. Пишем блок вывода данных для первого задания. Выводим на экран шифротекст, полученный однократным гаммированием исходной строки сгенерированным ключом (иллюстр. [-@fig:003])

Задание №1

```
In [3]:
print("Полученный при открытом ключе и тексте шифротекст:\n", encrypted_string)
print("В шестнадцатеричном виде:\n", hexadecimal_form(encrypted_string))
```

```
Полученный при открытом ключе и тексте шифротекст:
 іnѤєтЅљYщђќѰѽ}љЎЩрЉХq
В шестнадцатеричном виде:
 456 6e 464 454 442 405 459 59 449 452 45c 470 47d 7d 15 459 40e 429 440 409 425 71
```

Задание №1

4. Пишем блок вывода данных для второго задания. Выводим на экран случайный сгенерированный ключ, позволяющий вычислить один из вариантов расфишровки шифротекста, непосредственно сам вариант расшифровки, далее из шифротекста и исходной строки вычисляем и выводим исходный ключ и с его помощью выводим расшифрованный шифротекст (иллюстр. [-@fig:004])

Задание №2

```
In [4]: 1 print("Ключ, преобразовывающий шифротекст в один из возможных вариантов:\n", new_key)
        2 print("Один из вариантов прочтения открытого текста:\n", unencrypted_new_key)
        3
        4 print("\nИсходный ключ:\n", initial_key)
        5 print("Расшифрованный исходным ключом шифротекст:\n ", unencrypted_initial_key)
```

```
Ключ, преобразовывающий шифротекст в один из возможных вариантов:
 nKLteDoQmvnKcEUDZJVclD
Один из вариантов прочтения открытого текста:
 иЖШРЧсФФвлОВ@НєЬЖЖщ5

Исходный ключ:
 wNyjpNeyZlhNAQ5mNjwEjP
Расшифрованный исходным ключом шифротекст:
  С Новым Годом, друзья!
```

Задание №2

Выводы

В ходе работы мы успешно на практике освоили применение режима однократного гаммирования.

Ответы на контрольные вопросы

1. Поясните смысл однократного гаммирования.

- Гаммирование – это наложение (снятие) на открытые (зашифрованные) данные криптографической гаммы, то есть последовательности элементов данных, вырабатываемых с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Однократное гаммирование – это когда каждый символ попарно с символом ключа складываются по модулю 2 (XOR) (обозначается знаком $\oplus$).

Список литературы

1. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М. : Триумф, 2002. – 816 с.
2. Харин, Ю.С. Математические и компьютерные основы криптологии : учебное пособие / Ю.С. Харин, В.И. Берник, Г.В. Матвеев, С.В. Агиевич. – Мн. : Новое знание, 2003. – 382 с.
3. Д. С. Кулябов, А. В. Королькова, М. Н. Геворкян. Информационная безопасность компьютерных сетей: лабораторные работы. // Факультет физико-