

Лабораторная работа №7

Элементы криптографии. Однократное гаммирование

Топонен Никита Андреевич

Содержание

Цель работы	5
Задание	6
Теоретическое введение	7
Выполнение лабораторной работы	8
Ответы на контрольные вопросы	11
Выводы	13
Список литературы	14

Список иллюстраций

1	Вывод функции encryption	10
2	Вывод функции decryption	10
3	Проверка полученного ключа	10

Список таблиц

Цель работы

1. Освоить на практике применение режима однократного гаммирования

Задание

Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования.

Приложение должно:

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.

Теоретическое введение

Для выполнения данной лабораторной нет специальной теории. Необходимы общие знания в области компьютерных наук.

Выполнение лабораторной работы

```
import numpy as np
import operator as op
import sys

s = "С Новым Годом, друзья!"

def encryption(text):
    print("Открытый текст: ", text)

    new_text = []
    for i in text:
        new_text.append(i.encode("cp1251").hex())
    print("\nОткрытый текст в 16-ой системе: ", new_text)

    r = np.random.randint(0, 255, len(text))
    key = [hex(i)[2:] for i in r]

    new_key = []
    for i in key:
        new_key.append(i.encode("cp1251").hex().upper())
    print("\nКлюч в 16-ой системе: ", key)
```



```

xor_text = []
for i in range(len(new_text)):
    xor_text.append("{:02x}".format(int(key[i], 16) ^ int(new_text[i], 16)))
print("\nШифротекст в 16-ой системе: ", xor_text)

en_text = bytearray.fromhex("".join(xor_text)).decode("cp1251")
print("\nШифротекст: ", en_text)

return key, xor_text, en_text

def find_key(text, en_text):
    print("Открытый текст: ", text)
    print("\nШифротекст: ", en_text)

    new_text = []
    for i in text:
        new_text.append(i.encode("cp1251").hex())
    print("\nОткрытый текст в 16-ой системе: ", new_text)

    tmp_text = []
    for i in en_text:
        tmp_text.append(i.encode("cp1251").hex())
    print("\nШифротекст в 16-ой системе: ", tmp_text)

    xor_text = [hex(int(k, 16) ^ int(t, 16))[2:] for (k, t) in zip(new_text, tmp_text)]
    print("\nНайденный ключ в 16-ой системе: ", xor_text)
    return xor_text

k, t, et = encryption(s)

```

```
key = find_key(s, et)
```

```
if k == key:
    print("Ключ найден верно")
else:
    print("Ключ найден неверно")
```

Результаты работы программы:

```
Открытый текст: С Новым Годом, друзья!
Открытый текст в 16-ой системе: ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']
Ключ в 16-ой системе: ['5b', '54', '50', 'e0', '5', 'b', '9d', 'c8', 'e4', '28', '23', '3d', '1', '1b', 'ee', '86', '6b', '49', '7b', '7e', 'b7', '52']
Шифротекст в 16-ой системе: ['8a', '74', '9d', '0e', 'e7', 'f0', '71', 'e8', '27', 'c6', 'c7', 'd3', 'ed', '37', 'ce', '62', '9b', 'ba', '9c', '82', '48', '73']
Шифротекст: АtkВардиЖЭУн70b>ев,Нs
```

Рис. 1: Вывод функции encryption

```
Открытый текст: С Новым Годом, друзья!
Шифротекст: АtkВардиЖЭУн70b>ев,Нs
Открытый текст в 16-ой системе: ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']
Шифротекст текст в 16-ой системе: ['8a', '74', '9d', '0e', 'e7', 'f0', '71', 'e8', '27', 'c6', 'c7', 'd3', 'ed', '37', 'ce', '62', '9b', 'ba', '9c', '82', '48', '73']
Найденный ключ в 16-ой системе: ['5b', '54', '50', 'e0', '5', 'b', '9d', 'c8', 'e4', '28', '23', '3d', '1', '1b', 'ee', '86', '6b', '49', '7b', '7e', 'b7', '52']
```

Рис. 2: Вывод функции decryption

```
[22] 1 if k == key:
      2     print("Ключ найден верно")
      3 else:
      4     print("Ключ наадйен неверно")

Ключ найден верно
```

Рис. 3: Проверка полученного ключа

Ответы на контрольные вопросы

1. Поясните смысл однократного гаммирования. Гаммирование – выполнение операции XOR между элементами гаммы и элементами подлежащего сокрытию текста. Если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте.
2. Перечислите недостатки однократного гаммирования. Абсолютная стойкость шифра доказана только для случая, когда однократно используемый ключ, длиной, равной длине исходного сообщения, является фрагментом истинно случайной двоичной последовательности с равномерным законом распределения.
3. Перечислите преимущества однократного гаммирования. Во-первых, такой способ симметричен, т.е. двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение. Во-вторых, шифрование и расшифрование может быть выполнено одной и той же программой. Наконец, Криптоалгоритм не даёт никакой информации об открытом тексте: при известном зашифрованном сообщении C все различные ключевые последовательности K возможны и равновероятны, а значит, возможны и любые сообщения P .
4. Почему длина открытого текста должна совпадать с длиной ключа? Если ключ короче текста, то операция XOR будет применена не ко всем элементам

и конец сообщения будет не закодирован. Если ключ будет длиннее, то появится неоднозначность декодирования.

5. Какая операция используется в режиме однократного гаммирования, назовите её особенности? Наложение гаммы по сути представляет собой выполнение побитовой операции сложения по модулю 2, т.е. мы должны сложить каждый элемент гаммы с соответствующим элементом ключа. Данная операция является симметричной, так как прибавление одной и той же величины по модулю 2 восстанавливает исходное значение.
6. Как по открытому тексту и ключу получить шифротекст? В таком случае задача сводится к правилу: $C_i = P_i \oplus K_i$, т.е. мы поэлементно получаем символы зашифрованного сообщения, применяя операцию исключающего или к соответствующим элементам ключа и открытого текста.
7. Как по открытому тексту и шифротексту получить ключ? Подобная задача решается путем применения операции исключающего или к последовательностям символов зашифрованного и открытого сообщений: $K_i = P_i \oplus C_i$.
8. В чем заключаются необходимые и достаточные условия абсолютной стойкости шифра? Необходимые и достаточные условия абсолютной стойкости шифра:
 - полная случайность ключа;
 - равенство длин ключа и открытого текста;
 - однократное использование ключа.

Выводы

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

Список литературы

- Кулябов Д. С., Королькова А. В., Геворкян М. Н Лабораторная работа №7